SPRINGER
REFERENCE

Shashi Shekhar
Hui Xiong
Editors

Encyclopedia of GIS

Springer

Encyclopedia of GIS

Shashi Shekar • Hui Xiong (Eds.)

Encyclopedia of GIS

With 723 Figures and 90 Tables

Springer

Shashi Shekhar
McKnight Distinguished University Professor
Faculty of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
USA

Hui Xiong
Assistant Professor
Management Science and Information Systems Department
Rutgers Business School
Newark and New Brunswick Rutgers, the State University of New Jersey
Newark, NJ 07102
USA

Library of Congress Control Number: 2007933825

ISBN: 978-0-387-35973-1

This publication is available also as:
Print publication under ISBN: 978-0-387-30858-6 and
Print and electronic bundle under ISBN: 978-0-387-35975-5

springer.com

Printed on acid free paper SPIN: 11420118 2109letex – 5 4 3 2 1 0

To our families . . .

Foreword by Brian Berry

The publication of a definitive *Encyclopedia of GIS* that lays out many of the computer science/mathematics foundations of the field is a major event, the culmination of a half century of development. I was part of the earliest stirrings in the mid-1950s. A small group of geography graduate students at the University of Washington, William Garrison's "space cadets," began to assemble what became the foundations of contemporary spatial analysis and to refocus mathematical cartography while working with civil engineer Edgar Horwood on his attempts to use the printers of the time to produce grey-shaded maps. Our attention was captured by Sputnik, however, we didn't anticipate that the US's response, the rapid development of NASA and satellite systems, would be the key to the equally rapid development of remote sensing, or that global positioning would rewrite cartography. Among the innovations of the time were Torsten Hägerstrand's first simulation models of space-time diffusion processes and early econometric interest in spatial autocorrelation. Both themes are now central to spatial analysis.

The GIS focus shifted when Garrison, Marble and I relocated to the Chicago region. Garrison and I helped fund civil engineer Howard Fisher's first generation computer graphics software, SYMAP I, and Marble and I organized NSF workshops to spread the word and drew together an initial overview of the field in *Spatial Analysis*. Fisher took his ideas to the Ford Foundation and a subsequent grant to Harvard University, where he established the Laboratory for Computer Graphics. The Lab served as the focus for research in the field well into the 1970s, providing the spark to such innovators as Jack Dangermond, who subsequently established ESRI and created what became the world's most widely used computer graphics software. Meanwhile, hardware development proceeded apace, as did imaging and positioning capabilities created by the Department of Defense and NASA, facilitating the resulting emergence of digital cartography and the establishment of the first large-scale Geographic Information Systems such as the Canada Land Inventory. The rest, as they say, is history – albeit given a new dynamic by the Internet and the continued evolution of computing capabilities both on the desktop and in the supercomputer.

Fifty years after these beginnings, the result is a large and complex field spanning many disciplines, continuing to grow and to expand into an expanding array of applications. Cartographers have eschewed their pen-and-ink, and rudimentary mapmaking is at the fingertips of everyone with Internet access. Road atlases are fast giving way to satellite navigation systems. Congress continues to be concerned with the privacy issues raised by geographic information system capabilities, yet police and fire departments can no longer function effectively without GIS and Homeland Security without modern database and data mining capabilities. From city planning to store location, property taxation to highway building, disaster response to environmental management, there are few arenas in which GIS is not playing a significant role. What is important is the cross-cutting capability that was recognized when the NSF funded the Center for Spatially-Integrated Social Science (CSICC) at the University of California, Santa Barbara, or my own university's Ph.D. program, a joint venture of the School of Economic, Political and Policy Sciences, the School of Natural Sciences and Mathematics, and the School of Engineering and Computer Sciences.

I like to tell my colleagues that there are three levels of GIS education: "Driver's Ed," "Mr. Goodwrench," and "Design Team." Driver's Ed provides essential skills to the broad base of software users, the Mr. Goodwrenches of the world learn how to put together and maintain working software-hardware installations, while Design Teams create new and improved GIS capabilities. Most of the new data handling capabilities reside in the arenas of computer science/mathematics, while advances in inference are coming from innovations in the ability to handle space-time dynamics while simultaneously accounting for serial and spatial dependence.

The *Encyclopedia of GIS* provides an essential reference work for all three categories of users. In contrast to the current textbooks in the field, which are keyed to Driver's Ed, the *Encyclopedia* also provides a handy sourcebook for the Mr. Goodwrenches while defining the platform on which future Design Teams will build by focusing – more than existing works – on the computer science/mathematical foundations of the field. I know that the GIS community will value this important reference and guide, and I will cherish it as a milestone. It marks how far we have come – far beyond what our group of pioneers dreamed might be possible a half century ago. Professors Shekhar and Xiong and the considerable community of contributors to the collection have provided us with a comprehensive and authoritative treatment of the field, extensively cross-referenced with key citations and further readings. Importantly, it is available in both printed and XML online editions, the latter with hyperlinked citations. The GIS world will be indebted to them. No GIS bookshelf should, as they say, be without it.

McKinney, Texas
August, 2007

Brian J. L. Berry
School of Economic, Political and Policy Sciences
The University of Texas, Dallas
Dallas, TX
USA

Foreword by Michael Goodchild

Geographic Information Systems date from the 1960s, when computers were mostly seen as devices for massive computation. Very significant technical problems had to be solved in those early days: how did one convert the contents of a paper map to digital form (by building an optical scanner from scratch); how did one store the result on magnetic tape (in the form of a linear sequence of records representing the geometry of each boundary line as sequences of vertices); and how did one compute the areas of patches (using an elegant algorithm involving trapezia). Most of the early research was about algorithms, data structures, and indexing schemes, and, thus, had strong links to emerging research agendas in computer science.

Over the years, however, the research agenda of GIS expanded away from computer science. Many of the technical problems of computation were solved, and attention shifted to issues of data quality and uncertainty; the cognitive principles of user interface design; the costs and benefits of GIS; and the social impacts of the technology. Academic computer scientists interested in GIS wondered if their research would be regarded by their colleagues as peripheral – a marginally interesting application – threatening their chances of getting tenure. Repeated efforts were made to have GIS recognized as an ACM Special Interest Group, without success, though the ACM GIS conferences continue to attract excellent research.

The entries in this encyclopedia should finally lay any lingering doubts to rest about the central role of computer science in GIS. Some research areas, such as spatiotemporal databases, have continued to grow in importance because of the fundamental problems of computer science that they address, and are the subject of several respected conference series. Geospatial data mining has attracted significant attention from computer scientists as well as spatial statisticians, and it is clear that the acquisition, storage, manipulation, and visualization of geospatial data are special, requiring substantially different approaches and assumptions from those in other domains.

At the same time, GIS has grown to become a very significant application of computing. Sometime around 1995, the earlier view of GIS as an assistant performing tasks that the user found too difficult, complex, tedious, or expensive to do by hand, was replaced by one in which GIS became the means by which humans communicate what they know about the surface of Earth, with which they collectively make decisions about the management of land, and by which they explore the effects of alternative plans. A host of new issues suddenly became important: how to support processes of search, assessment, and retrieval of geospatial data; how to overcome lack of interoperability between systems; how to manage large networks of fixed or mobile sensors providing flows of real-time geographic data; how to offer useful services on the very limited platform of a cell phone; and how to adapt and evolve the technology in order to respond to emergencies and to provide useful intelligence. A revitalized research agenda for computer science emerged that shows no sign of diminishing, and is reflected in many of the topics addressed in this encyclopedia.

For example, computer scientists are engaged in the development of data structures, algorithms, and indexing schemes to support the hugely popular virtual globes (Google Earth, Microsoft's Virtual Earth, NASA's World Winds) that have emerged in the past few years and are stimulating a whole new generation of applications of geospatial technology. Research is ongoing on sensor networks, and the complex protocols that are needed to handle flows of real-time data from massive numbers of devices distributed over the Earth's surface, in areas of scientific interest such as the sea floor, in vehicles acquiring data on traffic movement, and in battlefields. Semantic interoperability, or the ability of systems to share not only data but the meaning of data, remains a thorny problem that will challenge the research community for many years to come.

As a collection of well-written articles on this expanding field, this encyclopedia is a welcomed addition to the GIS bookshelf. The fact that its compilers have chosen to emphasize the links between GIS and computer science is especially welcome. GIS is in many ways a *boundary object*, to use a term common in the community of science historians: a field that has emerged between two existing and recognized fields, in this case computer science and geography, and which has slowly established its own identity. As it does so, contributions such as this will help to keep those links alive, and to ensure that GIS continues to attract the interest of leading researchers in computer science.

Michael F. Goodchild
National Center for Geographic Information and Analysis
and Department of Geography
University of California
Santa Barbara, CA
USA

Preface

Interest in Geographic Information Systems, Science, and Services (GIS) has tremendously grown in recent years in many different ways. Researchers from a variety of academic disciplines are using spatial thinking and GIS tools to develop spatially-explicit models. Broad public interest in this subject is being fuelled by popular applications like Google maps, personal navigation devices, MapQuest, etc. Web-based software developers are increasingly exploring "mash-ups" integrating different information sources to web-based maps, such as Google Earth and MS Virtual Earth. Therefore, there is a need to bring key GIS concepts and results to a diverse audience as current GIS literature (e. g., textbooks, journals, conference-proceedings, trade-books) largely caters to either GIS specialists or end-users of popular GIS software.

The GIS research community is enthusiastically embracing encyclopedias, i. e., collections of articles on numerous topics in a field, as a tool to bring key ideas from a field to a general audience. This is not only evident from the preparation of multiple encyclopedias of GIS in the 2007–2008 time-frame, but also from the generous time commitments from GIS researchers contributing to the encyclopedia projects as authors, field editors and reviewers. The concurrent development of multiple GIS encyclopedias helped us define a focus, given the multi-disciplinary nature of the field. This encyclopedia focuses on computational aspects of a variety of GIS concepts for a variety of reasons. First, computational advances are making GIS available to a wide variety of end-users, software developers and researchers. Second, many geo-spatial datasets are large and growing rapidly due to advances in cyber-infrastructure, including sensor and data management technologies. This will make computational issues even more critical in the coming years. Finally, computational technologies are advancing at a rapid pace, making new capabilities possible every few years. While the recent advances, e. g., Google Earth and Microsoft Virtual Earth, look impressive, we are likely to see even bigger advances due to growing computing power in the years to come.

Despite the focus on computational aspects of GIS, it was still challenging to narrow down the list of possible articles in order to explain the key software, datasets, and processes used by geographers and computational scientists. After all, our goal was to provide a comprehensive and authoritative treatment of GIS, providing easy access to the field. We reviewed the topics in calls-for-papers for conferences and journals along with the keywords in relevant books and model curricula. We also consulted various leaders in the GIS area. Additionally, we tried to balance the set of topics to reflect the interests of industry, government and academia. For example, topics such as GIS standards as well as examples of GIS software from industry and the public domain were included in the topic list. Similarly, we included topics like crime mapping, evacuation planning and location-based services. Naturally, the list includes a variety of academic topics ranging from "Spatial Cognition" to "Statistical Modeling."

The next major challenge was to identify suitable field editors and authors representing the world leaders in selected topic areas. Many researchers usually focus on conference and journal papers. Yet, we were pleasantly surprised by the generosity of time and expertise received from so many enthusiastic GIS colleagues for this encyclopedia project. Field editors went beyond their call of duty to identify and invite potential authors, and review contributions in order to ensure technical quality while working with Springer project managers to keep the project on schedule. This is a wonderful sign of the energy and collegiality of the computational GIS research community.

Springer's encyclopedia group provided thoughtful guidelines for the entries. Typical entries are 3000 words and provide balance among definition, scientific fundamentals, application domains, and future trends. Many include short definitional entries cross-referenced to related regular entries in order to discuss specific terms and concepts such as the Global Positioning System, Digital Elevation/Terrain Model, and Remote Sensing. Regular entries include key citations and a list of recommended reading regarding the literature, and (online) internal hyperlinks to definitional entries and current standards. This Encyclopedia is also simultaneously available as an XML online reference with hyperlinked citations, cross-references, four-color art, links to Web-based maps, and other interactive features.

Key Features at a Glance

- Immediate point of entry into the field for researchers
- A–Z format allows easy, intuitive access for newcomers to the field
- Many headers for easy skimming and navigation of topics
- Includes coverage of GIS standards in development by ISO
- Cross-referenced entries
- Internationally renowned editorial board, both scientifically and geographically diverse
- Hundreds of contributors ensure balanced coverage
- Interactive features add to convenience, ease of use and understanding
- Peer-reviewed entries assure researchers that information is vetted
- eReference is available at springerlink.com

Content Organization

The encyclopedia is divided into 41 fields, each one an important sub-area within GIS. These fields include: Basic Concepts; Basic Storage and Retrieval Structure; Cartography and Visualization; Commercial GIS; Commercial Spatial Databases; Critical Evaluation of Standard Proposals; Data Exchange and Interoperability; Digital Road Map; Emergency Evacuations; Evacuation Planning and Operations; Geosensor Networks; GeoSpatial Semantic Web; GIS in Business Intelligence, Routing; GIS Issues and Applications; Indoor Positioning; Information Collection Using Sensor Network; Open Source GIS Software; Photogrammetry; Representation of Inexact Spatial Information; Road Network Databases; Security and Privacy in Geospatial Information Systems; Spatial Analysis; Spatial Aspects of Bioinformatics; Spatial Aspects of Distributed Computing; Spatial Aspects of Mobile Computing; Spatial Association Discovery; Spatial Colocation Rule Mining; Spatial Constraint Databases; Spatial Data Warehousing and Decision Support; Spatial Database Modeling for Applications; Spatial Indexing; Spatial Outlier Detection; Spatial Prediction; Spatial Thinking; Spatial Time Series; Spatial Uncertainty and Imprecision; Spatio-Temporal Data Modeling; Spatio-Temporal Databases; Statistical Modeling for Spatial Data; Tesselation Data Models; and Use of Spatial Data for Simulation.

Acknowledgements

Many people have played a part in the production of this book and we are extremely grateful to them. In particular, field editors who have made excellent efforts include Vijay Atluri, Sudipto Banerjee, Yvan Bedard, Sanjay Chawla, Robert Denaro, Liping Di, Frederico Fonseca, Andrew Frank, Oscar Franzese, Dimitrios Gunopulos, Erik Hoel, Kathleen Hornsby, Yan Huang, Robert Kauffman, Baris Kazar, Sangho Kim, Ravi Kothuri, Phaedon Kyriakidis, Xinrong Li, Henry Liu, Chang-Tien Lu, Nikos Mamoulis, Helmut Mayer, Liqiu Meng, Mohamed Mokbel, Andreas Neumann, Silvia Nittel, Leon Osborne, Sudhanshu Sekhar Panda, Srinivasan Parthasarathy, Peter Revesz, Ashok Samal, Markus Schneider, Cyrus Shahabi, Jayant Sharma, Yufei Tao, Vassilis Tsotras, Ouri Wolfson, Chaowei Yang, Pusheng Zhang, and Naijun Zhou.

We would like to thank the members of the spatial database research group in the Computer Science Department at the University of Minnesota. They contributed in many different ways including providing literature surveys, organizing topics, and reviewing articles. We would also like to thank some of the students enrolled in Csci 8701 and Csci 8715 for contributing articles.

Finally, special thanks are due to many people at Springer for their enthusiasm, advice, and support. In particular, we would like to thank Susan Lagerstrom-Fife, Oona Schmid, Jennifer Carlson, Andrea Schmidt, Simone Tavenrath, Sharon Palleschi, and Yana Lambert, who at different times have played key roles in the development of this book.

August, 2007 Shashi Shekhar and Hui Xiong

Topical Table of Contents

Fonseca, Frederico: GeoSpatial Semantic Web

Frank, Andrew: Spatial Thinking

Franzese, Oscar: Emergency Evacuations

Gunopulos, Dimitrios: Information Collection Using Sensor Network

Hoel, Erik: Commercial GIS

Hornsby, Kathleen: Spatio-temporal Data Modeling

Huang, Yan: Spatial Co-location Rule Mining

Kauffman, Robert: GIS Issues and Applications

Kazar, Baris: Spatial Prediction

Kim, Sangho: Basic Concepts

Kothuri, Ravi: Commercial Spatial Databases

Kyriakidis, Phaedon: Spatial Uncertainty and Imprecision

Li, Xinrong: Indoor Positioning

Liu, Henry: Evacuation Planning and Operations

Lu, Chang-Tien: Spatial Outlier Detection

Mamoulis, Nikos: Spatial Data Warehousing and Decision Support

Mayer, Helmut: Photogrammetry

Meng, Liqui: Cartography and Visualization

Mokbel, Mohamed: Basic Storage and Retrieval Structure

Neumann, Andreas: Open Source GIS Software

Nittel, Silvia: Geosensor Networks

Osborne, Jr., Leon : Use of Spatial Data for Simulation

Panda, Sudhanshu: Tesselation Data Models

Parthasarathy, Srinivasan: Spatial Aspects of Bioinformatics

Revesz, Peter: Spatial Constraint Databases

Samal, Ashok: Spatial Analysis

Schneider, Markus: Representation of Inexact Spatial Information

Shahabi, Cyrus: Road Network Databases

Sharma, Jayant: GIS in Business Intelligence, Routing

Tao, Yufei: Spatial Indexing

Tsotras, Vassilis: Spatio-temporal Databases

Wolfson, Ouri: Spatial Aspects of Mobile Computing

Yang, Chaowei (Phil): Spatial Aspects of Distributed Computing

Zhang, Pusheng: Spatial Time Series

Zhou, Naijun: Data Exchange and Interoperability

About the Editors

Shashi Shekhar
McKnight Distinguished University Professor
Faculty of Computer Science and Engineering
University of Minnesota
Minneapolis, MN
USA

shekhar@cs.umn.edu
http://www.cs.umn.edu/~shekhar

Professor Shekhar was elected as an IEEE Fellow and received the IEEE Technical Achievement Award for contributions to spatial database storage methods, data mining, and Geographic Information Systems (GIS). He has a distinguished academic record that includes 200+ refereed papers and two books including a textbook on Spatial Databases (Prentice Hall, 2003). He is serving as a member of the mapping science committee of the NRC/NAS (National Research Council National Academy of Sciences) (2004-2009), and the steering committee of the ACM Symposium on GIS. He is also serving as a co-Editor-in-Chief of Geo-Informatica: An International Journal on Advances of Computer Science for GIS. He has served as a member of the NRC/NAS Committee, reviewing basic and applied research at the National Geo-spatial-Intelligence Agency (NGA), the Board of Directors of the University Consortium on GIS (2003-2004), the editorial boards of IEEE Transactions on Knowledge and Data Engineering as well as the IEEE-CS Computer Science & Engineering Practice Board. He also served as a program co-chair for the ACM International Workshop on Advances in GIS (1996).

Professor Shekhar is a leading researcher in the area of spatial databases and spatial data mining, an interdisciplinary area at the intersection of Computer Science and GIS. A major goal of his research is to understand the computational structure of very large spatial computations (e. g., data analysis via spatial querying and spatial data mining) needed by social and physical sciences as well as engineering disciplines. Earlier, his research developed core technologies behind in-vehicle navigation devices as well as web-based routing services, revolutionizing outdoor navigation in the urban environment in the last decade. His research results are now playing a critical role in evacuation route planning for homeland security and were recognized via the CTS partnership award (2006) for significantly impacting transportation.

Professor Shekhar's general area of research includes data and knowledge engineering with a focus on storage, management and analysis of scientific and geographic data, information and knowledge. Major contributions in data engineering and database systems include the Connectivity-Clustered Access Method (CCAM), a new storage and access method for spatial networks which outperforms traditional indexes (e. g., R-tree family) in carrying out network computations. Other contributions are related to semantic query optimization and high performance geographic databases. In knowledge engineering, his work focuses on spatial data mining and neural networks. Major contributions include the notion of "co-location" patterns in spatial datasets, characterization of the computational structure of "spatial outlier" detection, faster algorithms for estimating parameters for the spatial auto-regression model as well one of the most scalable parallel formulations of the back-propagation learning algorithms for neural networks.

Hui Xiong
Assistant Professor
Management Science and Information Systems Department
Rutgers Business School
Newark and New Brunswick, Rutgers, the State University of New Jersey
Newark, NJ
USA

hxiong@andromeda.rutgers.edu
http://datamining.rutgers.edu

Professor Xiong received his Ph.D. in computer science from the University of Minnesota in 2005. He is currently an assistant professor in the Management Science and Information Systems Department at Rutgers, the State University of New Jersey. His general area of research is data and knowledge engineering, with a focus on developing effective and efficient data analysis techniques for emerging data intensive applications. He has published prolifically in refereed journals and conference proceedings, such as IEEE Transactions on Knowledge and Data Engineering, the VLDB Journal, the Data Mining and Knowledge Discovery Journal, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), SIAM International Conference on Data Mining (SDM), IEEE International Conference on Data Mining (ICDM), and ACM International Symposium on Advances in Geographic Information Systems (ACM GIS). He is the co-editor of Clustering and Information Retrieval (Kluwer Academic Publishers, 2003) and the author of Hyperclique Pattern Discovery: Algorithms and Applications (ProQuest Information and Learning, 2006). He has served regularly in the organization committees and the program committees of a number of international conferences and workshops. Furthermore, he has been a reviewer for the leading academic journals in his fields. He is a member of the IEEE Computer Society, the Sigma Xi, and the ACM. Professor Xiong was the recipient of the Junior Faculty Teaching Excellence Award at the Rutgers Business School in 2007.

Professor Xiong is a well-known researcher in the field of data mining. A major goal of his research is to understand the computational structure of very large computations needed in science, business, and engineering disciplines. His major research contributions include the notion of hyperclique patterns, characterization of the computational structure of the phi correlation coefficient, enhancing data analysis with noise removal, a systematic study of clustering validation measures from a data distribution perspective, and the development of local decomposition for rare class analysis.

Field Editors

ATLURI, VIJAY
Rutgers University
Newark, NJ
USA

BANERJEE, SUDIPTO
University of Minnesota
Minneapolis, MN
USA

BEDARD, YVAN
Laval University
Quebec City, QC
Canada

CHAWLA, SANJAY
The University of Sydney
Sydney, NSW
Australia

DENARO, ROBERT
NAVTEQ
Chicago, IL
USA

DI, LIPING
George Mason University
Fairfax, VA
USA

FONSECA, FREDERICO
The Pennsylvania State University
University Park, PA
USA

FRANK, ANDREW
Technical University of Vienna
Vienna
Austria

FRANZESE, OSCAR
Oak Ridge National Laboratory
Oak Ridge, TN
USA

GUNOPULOS, DIMITRIOS
University of California, Riverside
Riverside, CA
USA

HOEL, ERIK
ESRI
Redlands, CA
USA

HORNSBY, KATHLEEN
The University of Iowa
Iowa City, IA
USA

HUANG, YAN
University of North Texas
Denton, TX
USA

KAUFFMAN, ROBERT
Arizona State University, Tempe, AZ
University of Minnesota, Minneapolis, MN
USA

KAZAR, BARIS
Oracle Corporation
Nashua, NH
USA

KIM, SANGHO
ESRI
Redlands, CA
USA

KOTHURI, RAVI
Oracle Corporation
Nashua, NH
USA

KYRIAKIDIS, PHAEDON
University of California, Santa Barbara
Santa Barbara, CA
USA

List of Contributors

ACHARYA, RAJ
The Pennsylvania State University
University Park, PA
USA

ADAM, NABIL R.
Rutgers University
Newark, NJ
USA

AGARWAL, PRAGYA
University College London
London
UK

AGGETT, GRAEME
Riverside Technology, Inc.
Fort Collins, CO
USA

ALAM, ASHRAFUL
The University of Texas-Dallas
Dallas, TX
USA

ALAVI, BARDIA
Worcester Polytechnic Institute
Worcester, MA
USA

ALTHEIDE, PHYLLIS
U.S. Geological Survey
Rolla, MO
USA

AMES, DANIEL P.
Idaho State University
Pocatello, ID
USA

ANSELMO, ALLEN
Idaho State University
Pocatello, ID
USA

ARAB, ALI
University of Missouri-Columbia
Columbia, MO
USA

AREF, WALID G.
Purdue University
West Lafayette, IN
USA

ASTROTH, JOE
Autodesk Location Services
San Rafael, CA
USA

ATLURI, VIJAY
Rutgers University
Newark, NJ
USA

BADARD, THIERRY
Laval University
Quebec, QC
Canada

BAI, YUQI
George Mason University
Fairfax, VA
USA

BAILEY-KELLOGG, CHRIS
Dartmouth College
Hanover, NH
USA

BAKIRAS, SPIRIDON
University of Hong Kong
Hong Kong
China

BALRAM, SHIVANAND
Simon Fraser University
Burnaby, B.C.
Canada

BANERJEE, SUDIPTO
University of Minnesota
Mineapolis, MN
USA

BARRETT, CHRISTOPHER L.
Virginia Tech
Blacksburg, VA
USA

BATTY, PETER
Intergraph Corporation
Huntsville, AL
USA

BERCHTOLD, STEFAN
stb GmbH
Augsburg
Germany

BETTINI, CLAUDIO
University of Milan
Milan
Italy

BHADURI, BUDHENDRA
Oak Ridge National Laboratory
Oak Ridge, TN
USA

BISHR, YASER
Image Matters LLC
Leesburg, VA
USA

BITTNER, THOMAS
State University of New York at Buffalo
Buffalo, NY
USA

BIVAND, ROGER
Norwegian School of Economics and Business Administration
Bergen
Norway

BOEDIHARDJO, ARNOLD P.
Virginia Polytechnic Institute and State University
Falls Church, Virginia
USA

BRAGA, JOSÉ LUÍS
Federal University of Vicosa
Viçosa
Brazil

BRAKATSOULAS, SOTIRIS
RA Computer Technology Institute
Athens
Greece

BRODEUR, JEAN
Natural Resources Canada
Sherbrooke, QC
Canada

BROWN, JAMES D.
National Weather Service
Silver Spring, MD
USA

BRUNS, DALE A.
Wilkes University
Wilkes-Barre, PA
USA

BULLIMORE, MARK A.
The Ohio State University
Columbus, OH
USA

BURGHARDT, DIRK
University of Zurich
Zurich
Switzerland

BURK, THOMAS E.
University of Minnesota
St. Paul, MN
USA

BUSTOS, BENJAMIN
University of Chile
Santiago
Chile

BÉDARD, YVAN
Laval University
Quebec City, QC
Canada

BÖHM, CHRISTIAN
University of Munich
Munich
Germany

CAMPOS, JORGE
Salvador University – UNIFACS
Salvador
Brazil

CARON, CLAUDE
Sherbrooke University
Sherbrooke, QC
Canada

CHAKHAR, SALEM
LAMSADE
University of Paris Dauphine
Paris
France

CHAMBERLAND-TREMBLAY, DANIEL
Sherbrooke University
Sherbrooke, QC
Canada

CHAUDHRY, OMAIR
The University of Edinburgh
Edinburgh
UK

CHAWLA, SANJAY
The University of Sydney
Sydney, NSW
Australia

CHEN, AIJUN
George Mason University
Fairfax, VA
USA

CHEN, CHING-CHIEN
Geosemble Technologies
El Segundo, CA
USA

CHEN, CINDY X.
University of Massachusetts-Lowell
Lowell, MA
USA

CHEN, FENG
Virginia Polytechnic Institute and State University
Falls Church, VA
USA

CHEN, LAILIN
Idaho State University
Pocatello, ID
USA

CHENG, DIHAN
Boston University
Boston, MA
USA

CHENG, REYNOLD
Hong Kong Polytechnic University
Hong Kong
China

CHIU, YI-CHANG
University of Arizona
Tucson, AZ
USA

CHOW, CHI-YIN
University of Minnesota
Minneapolis, MN
USA

CHRISTAKOS, GEORGE
San Diego State University
San Diego, CA
USA

CHUN, SOON AE
College of Staten Island – City University of New York
New York, NY
USA

CLEMENTINI, ELISEO
University of L'Aquila
L'Aquila
Italy

COHN, ANTHONY G.
University of Leeds
Leeds
UK

CRAMER, MICHAEL
University of Stuttgart
Stuttgart
Germany

CROSSLAND, MARTIN D.
Oral Roberts University
Tulsa, OK
USA

DAI, JING
Virginia Polytechnic Institute and State University
Falls Church, VA
USA

DANKO, DAVID M.
Environmental Systems Research Institute, Inc
Vienna, VA
USA

DAVIS JR., CLODOVEU A.
Pontifical Catholic University of Minas Gerais
Belo Horizonte
Brazil

DI, LIPING
George Mason University
Fairfax, VA
USA

DÖLLNER, JÜRGEN
University of Potsdam
Potsdam
Germany

DOS SANTOS JR., RAIMUNDO F.
Virginia Polytechnic Institute and State University
Herndon, VA
USA

DRAGIĆEVIĆ, SUZANA
Simon Fraser University
Burnaby, B.C.
Canada

DUBIN, ROBIN
Case Western Reserve University
Cleveland, OH
USA

DUCKHAM, MATT
University of Melbourne
Melbourne, VIC
Australia

DUNSFORD, HAROLD
Idaho State University
Pocatello, ID
USA

DÜZGÜN, H. SEBNEM
Middle East Technical University
Ankara
Turkey

ELBASSIONI, KHALED
Max-Planck-Institute for Computer Science
Saarbrücken
Germany

ELMASRY, AMR
Alexandria University
Alexandria
Egypt

EUBANK, STEPHEN
Virginia Tech
Blacksburg, VA
USA

EVANS, JOHN
Global Science & Technology Inc.
NASA Geosciences Interoperability Office
Greenbelt, MD
USA

FEEMAN, TIMOTHY G.
Villanova University
Villanova, PA
USA

FERHATOSMANOGLU, HAKAN
The Ohio State University
Columbus, OH
USA

FILBERT, KATIE M.
National Institute of Justice
Washington, D.C.
USA

FILHO, JUGURTA LISBOA
Federal University of Vicosa
Viçosa
Brazil

FINLEY, ANDREW O.
Michigan State University
East Lansing, MI
USA

FISHER, PETER
City University
London
UK

FOLEY, KRISTEN M.
North Carolina State University
Raleigh, NC
USA

FONSECA, FREDERICO
The Pennsylvania State University
University Park, PA
USA

FRANCICA, JOSEPH R.
Directions Media
Huntsville, AL
USA

FRANZESE, OSCAR
Oak Ridge National Laboratory
Oak Ridge, TN
USA

FUENTES, MONTSERRAT
North Carolina State University
Raleigh, NC
USA

GADDY, STEPHEN G.
Meridian Environmental Technology, Inc.
Grand Forks, ND
USA

GALTON, ANTONY
University of Exeter
Exeter
UK

GANDHI, VIJAY
University of Minnesota
Minneapolis, MN
USA

GANGODAGAMAGE, CHANDANA
University of Minnesota
Minneapolis, MN
USA

GEORGE, BETSY
University of Minnesota
Minneapolis, MN
USA

GHINITA, GABRIEL
National University of Singapore
Singapore

GIBAS, MICHAEL
The Ohio State University
Columbus, OH
USA

GOODCHILD, MICHAEL F.
University of California
Santa Barbara, CA
USA

GOOVAERTS, PIERRE
BioMedware Inc
Ann Arbor, MI
USA

GOSHEY, MICHAEL
University of Minnesota
Minneapolis, MN
USA

GRABOW, DAMON
University of North Dakota
Grand Forks, ND
USA

GRUTESER, MARCO
Rutgers University
Piscataway, NJ
USA

GUDMUNDSSON, JOACHIM
NICTA
Sydney, NSW
Australia

GUERRERO, IGNACIO
Intergraph Corporation
Huntsville, AL
USA

GUNOPULOS, DIMITRIOS
University of California – Riverside
Riverside, CA
USA

GUPTA, SANDEEP
University of California-Riverside
Riverside, CA
USA

GÜTING, RALF HARTMUT
Distance University of Hagen
Hagen
Germany

HADAYA, PIERRE
University of Quebec at Montreal
Montréal, QC
Canada

HADJIELEFTHERIOU, MARIOS
AT&T Labs Inc
Florham Park, NJ
USA

HALBEY-MARTIN, MALTE
Free University of Berlin
Berlin
Germany

HAN, JIAWEI
University of Illinois at Urbana-Champaign
Urbana, IL
USA

HART, ROBERT D.
Meridian Environmental Technology, Inc.
Grand Forks, ND
USA

HATAMI, AHMAD
Worcester Polytechnic Institute
Worcester, MA
USA

HEIPKE, CHRISTIAN
Leibniz University
Hannover
Germany

HELLWICH, OLAF
Berlin University of Technology
Berlin
Germany

HEUVELINK, GERARD B.M.
Wageningen University
Wageningen
The Netherlands

HIRTLE, STEPHEN C.
University of Pittsburgh
Pittsburgh, PA
USA

HOEL, ERIK
ESRI
Redlands, CA
USA

HOH, BAIK
Rutgers University
Piscataway, NJ
USA

HOOTEN, MEVIN B.
Utah State University
Logan, UT
USA

HORNING, NED
American Museum of Natural History
New York, NY
USA

HU, WEI
International Business Machines Corp.
Rochester, MN
USA

HUANG, YAN
University of North Texas
Denton, TX
USA

HUGENTOBLER, MARCO
ETH Zurich
Zurich
Switzerland

HURNI, LORENZ
ETH Zurich
Zurich
Switzerland

IOCHPE, CIRANO
Federal University of Rio Grande do Sul (UFRGS)
Porto Alegre
Brazil

JAGADISH, H.V.
University of Michigan
Ann Arbor, MI
USA

JANEJA, VANDANA P.
University of Maryland – Baltimore County
Baltimore, MD
USA

JENSEN, CHRISTIAN S.
Aalborg University
Aalborg
Denmark

JIN, GUANG
University of Maine – Orono
Orono, Maine
USA

JOLMA, ARI
Helsinki University of Technology
Espoo
Finland

KALASHNIKOV, DMITRI
Purdue University
West Lafayette, IN
USA

KALNIS, PANOS
National University of Singapore
Singapore

KALOGERAKI, VANA
University of California – Riverside
Riverside, CA
USA

KAMEL, IBRAHIM
University of Sharjah
Sharjah
UAE

KANAAN, MUZAFFER
Worcester Polytechnic Institute
Worcester, MA
USA

KANG, JAMES M.
University of Minnesota
Minneapolis, MN
USA

KANTARCIOGLU, MURAT
The University of Texas-Dallas
Dallas, TX
USA

KEENAN, PETER
University College Dublin
Dublin
Ireland

KEIM, DANIEL
University of Konstanz
Konstanz
Germany

KEMEÇ, SERKAN
Middle East Technical University
Ankara
Turkey

KEOGH, EAMONN
University of California – Riverside
Riverside, CA
USA

KHAN, LATIFUR
The University of Texas-Dallas
Dallas, TX
USA

KIM, SANGHO
ESRI
Redlands, CA
USA

KNOBLOCK, CRAIG A.
University of Southern California
Marina del Rey, CA
USA

KOLLIOS, GEORGE
Boston University
Boston, MA
USA

KOTHURI, RAVI
Oracle
Nashua, NH
USA

KOU, YUFENG
Virginia Tech
Falls Church, VA
USA

KRAAK, MENNO-JAN
ITC – International Institute for Geoinformation Science and Earth Observation
Enschede
The Netherlands

KRIEGEL, HANS-PETER
Ludwig-Maximilians-University
Munich
Germany

KUIJPERS, BART
Hasselt University and Transnational University of Limburg
Diepenbeek
Belgium

KULIK, LARS
The University of Melbourne
Melbourne, VIC
Australia

KUNATH, PETER
Ludwig-Maximilians-University
Munich
Germany

KUSHKI, A.
University of Toronto
Toronto, ONT
Canada

LACERDA ALVES, LEONARDO
Pontifical Catholic University of Minas Gerais
Belo Horizonte
Brazil

LANIER, DWIGHT
Gainesville State College
Gainesville, GA
USA

LAPIERRE, CÉDRIC
University of Sherbrooke
Sherbrooke, QC
Canada

LARRIVÉE, SUZIE
Laval University
Quebec City, QC
Canada

LAUBE, PATRICK
University of Melbourne
Melbourne
Australia

LAUDATI, ROBERT P.
GE Energy
Englewood, CO
USA

LAWSON, ANDREW B.
University of South Carolina
Columbia, SC
USA

LAZARIDIS, IOSIF
University of California
Irvine, CA
USA

LESAGE, JAMES P.
Texas State University – San Marcos
San Marcos, TX
USA

LEVINE, NED
Ned Levine & Associates
Houston, TX
USA

LEWIS, BRYAN
Virginia Tech
Blacksburg, VA
USA

LI, BIN
Central Michigan University
Mount Pleasant, MI
USA

LI, FEIFEI
Boston University
Boston, MA
USA

LI, LIXIN
Georgia Southern University
Statesboro, GA
USA

LI, WENWEN
George Mason University
Fairfax, VA
USA
Chinese Academy of Sciences
Beijing
China

LI, XINRONG
University of North Texas
Denton, TX
USA

LIBOUREL, THÉRÈSE
University of Montpellier II
Montpellier Cedex 5
France

LIM, HYEEUN
University of Minnesota
Minneapolis, MN
USA

LIME, STEVE
Minnesota Department of Natural Resources
St. Paul, MN
USA

LIN, DAN
Purdue University
West Lafayette, IN
USA

LIN, HENRY
The Pennsylvania State University
University Park, PA
USA

LIN, SONG
University of California
Riverside, CA
USA

LIU, CHENG
Oak Ridge National Laboratory
Oak Ridge, TN
USA

LIU, HENRY X.
University of Minnesota
Minneapolis, MN
USA

LU, CHANG-TIEN
Virginia Tech
Falls Church, VA
USA

LUHMANN, THOMAS
Institute for Applied Photogrammetry and Geoinformatics
Oldenburg
Germany

LUO, YAN
University of Illinois at Chicago
Chicago, IL
USA

LUPP, MARKUS
lat/lon GmbH
Bonn
Germany

MACKANESS, WILLIAM A.
The University of Edinburgh
Edinburgh
UK

MAGUIRE, DAVID J.
ESRI
Redlands, CA
USA

MAITRA, JULIE BINDER
Federal Geographic Data Committee (FGDC)
Reston, VA
USA

MAMOULIS, NIKOS
University of Hong Kong
Hong Kong
China

MANOLOPOULOS, YANNIS
Aristotle University
Thessaloniki
Greece

MARATHE, MADHAV V.
Virginia Tech
Blacksburg, VA
USA

MARSOLO, KEITH
The Ohio State University
Columbus, OH
USA

MASCETTI, SERGIO
University of Milan
Milan
Italy

MCCOLL, CHRIS
Central Washington University
Ellensburg, WA
USA

MCGLONE, J. CHRIS
SAIC
Chantilly, VA
USA

MEHROTRA, SHARAD
University of California – Irivne
Irvine, CA
USA

MEWES, JOHN J.
Meridian Environmental Technology, Inc.
Grand Forks, ND
USA

MICHAELIS, CHRISTOPHER
Idaho State University
Pocatello, ID
USA

MILLER, HARVEY J.
University of Utah
Salt Lake City, UT
USA

MIRALLES, ANDRÉ
Cemagref Agricultural and Environmental Engineering Research
Montpellier Cedex 5
France

MITASOVA, HELENA
North Carolina State University
Raleigh, NC
USA

MOKBEL, MOHAMED F.
University of Minnesota
Minneapolis, MN
USA

MOON, GEORGE
MapInfo Corporation
Troy, New York
USA

MORETZ, DAVID
University of Minnesota
Minneapolis, MN
USA

MOUSSEAU, VINCENT
LAMSADE
University of Paris Dauphine
Paris
France

MYERS, WAYNE
The Pennsylvania State University
University Park, PA
USA

NETELER, MARKUS
The Trentino Cultural Institute
Trento
Italy

NEUMANN, ANDREAS
ETH Zurich
Zurich
Switzerland

NITTEL, SILVIA
University of Maine – Orono
Orono, ME
USA

O'NEIL, D.J.
University of Minnesota
Minneapolis, MN
USA

OOI, BENG CHIN
National University of Singapore
Singapore

OSBORNE, LEON F.
Univerity of North Dakota
Meridian Environmental Technology, Inc.
Grand Forks, ND
USA

PACE, R.KELLEY
Louisiana State University
Baton Rouge, LA
USA

PAHLAVAN, KAVEH
Worcester Polytechnic Institute
Worcester, MA
USA

PANDA, SUDHANSHU SEKHAR
Gainesville State College
Gainesville, GA
USA

PAPADIAS, DIMITRIS
Hong Kong University of Science and Technology
Hong Kong
China

PARENT, CHRISTINE
University of Lausanne
Lausanne
Switzerland

PARTHASARATHY, SRINIVASAN
The Ohio State University
Columbus, OH
USA

PATIL, GANAPATI P.
The Pennsylvania State University
University Park, PA
USA

PATWARI, NEAL
University of Utah
Salt Lake City, UT
USA

PAULY, ALEJANDRO
University of Florida
Gainesville, FL
USA

PERCIVALL, GEORGE
Open Geospatial Consortium
Herndon, VA
USA

PFOSER, DIETER
RA Computer Technology Institute
Athens
Greece

PHOHA, SHASHI
The Pennsylvania State University
University Park, PA
USA

PICK, JAMES B.
University of Redlands
Redlands, CA
USA

PLATANIOTIS, K.N.
University of Toronto
Toronto, ONT
Canada

POTTINGER, RACHEL
University of British Columbia
Vancouver, BC
Canada

PRABHAKAR, SUNIL
Purdue University
West Lafayette, IN
USA

RACICOT, AARON
Ecotrust
Portland, OR
USA

RAMAKRISHNAN, NAREN
Virginia Tech
Blacksburg, VA
USA

RASKIN, ROBERT
California Institute of Technology
Pasadena, CA
USA

RAUBAL, MARTIN
University of California
Santa Barbara, CA
USA

RAVADA, SIVA
Oracle
Nashua, NH
USA

RAVISHANKAR, CHINYA V.
University of California-Riverside
Riverside, CA
USA

REICHENBACHER, TUMASCH
University of Zurich
Zurich
Switzerland

RENZ, MATTHIAS
Ludwig-Maximilians-University
Munich
Germany

REVESZ, PETER
University of Nebraska – Lincoln
Lincoln, NE
USA

ROCHE, STÉPHANE
Laval University
Québec, QC
Canada

ROTTENBACHER, CHRISTINE
Technical University of Vienna
Vienna
Austria

RUAS, ANNE
IGN-France
Saint-Mandé
France

RÖNSDORF, CARSTEN
Ordnance Survey
Southampton
UK

SAADA, MOKHTAR
University of Sherbrooke
Sherbrooke, QC
Canada

SALMENKIVI, MARKO
University of Helsinki
Helsinki
Finland

ŠALTENIS, SIMONAS
Aalborg University
Aalborg
Denmark

SAMAL, ASHOK
University of Nebraska-Lincoln
Lincoln, NE
USA

SANDERSON, MARK
University of Sheffield
Sheffield
UK

SANDERSON, MICHAEL
Laser-Scan
Cambridge
UK

SANKEY, TEMUULEN TSAGAAN
Idaho State University
Pocatello, ID
USA

SARJAKOSKI, L. TIINA
Finnish Geodetic Institute
Masala
Finland

SARJAKOSKI, TAPANI
Finnish Geodetic Institute
Masala
Finland

SARLITTO, DANIEL J.
SummitPoint Management LLC
Issaquah, WA
USA

SCHEUERMANN, PETER
Northwestern University
Evanston, IL
USA

SCHNEIDER, MARKUS
University of Florida
Gainesville, FL
USA

SEMERJIAN, CHRISTOPHER J.
Gainesville State College
Oakwood, Georgia
USA

SESTER, MONIKA
Leibniz University of Hannover
Hannover
Germany

SETH, SHARAD
University of Nebraska-Lincoln
Lincoln, NE
USA

SHA, LEI
Mayo Clinic
Rochester, MN
USA

SHAHABI, CYRUS
University of Southern California
Los Angeles, CA
USA

SHARIFZADEH, MEHDI
University of Southern California
Los Angeles, CA
USA

SHEKHAR, SHASHI
University of Minnesota
Minneapolis, MN
USA

SHIN, HEECHANG
Rutgers University
Newark, NJ
USA

SINHA, ABHINAYA
University of Minnesota
Minneapolis, MN
USA

SOH, LEEN-KIAT
University of Nebraska – Lincoln
Lincoln, NE
USA

SOHEILI, AMIR
ESRI
Redlands, CA
USA

SPACCAPIETRA, STEFANO
Federal Institute of Technology Lausanne
Lausanne
Switzerland

SPEIČYS, LAURYNAS
Aalborg University
Aalborg
Denmark

SPERBER, MARCI
University of Minnesota
Minneapolis, MN
USA

SRINIVASAN, SUMEETA
Harvard University
Cambridge, MA
USA

STELL, JOHN G.
University of Leeds
Leeds
UK

STROBL, CHRISTIAN
German Remote Sensing Data Center (DFD)
German Aerospace Center (DLR)
Weßling
Germany

STUBKJÆR, ERIK
Aalborg University
Aalborg
Denmark

SUBBIAH, GANESH
The University of Texas-Dallas
Dallas, TX
USA

SUTTON, TIM
Reference Center on Environmental Information
Campinas
Brazil

TAO, YUFEI
Chinese University of Hong Kong
Hong Kong
China

TARANTILIS, CHRISTOS D.
Athens University of Economics & Business
Athens
Greece

TENG, SHANG-HUA
Boston University
Boston, MA
USA

THAKUR, NILU
University of Minnesota
Minneapolis, MN
USA

THEODORIDIS, YANNIS
University of Piraeus
Piraeus
Greece

THURAISINGHAM, BHAVANI
The University of Texas-Dallas
Dallas, TX
USA

THÉAU, JÉRÔME
Idaho State University
Pocatello, ID
USA

TIMPF, SABINE
University of Würzburg
Würzburg
Germany

TRAJCEVSKI, GOCE
Northwestern University
Evanston, IL
USA

TROY, AUSTIN
University of Vermont
Burlington, VT
USA

TRYFONA, NECTARIA
Talent Information Systems SA
Athens
Greece

TSOTRAS, VASSILIS
University of California-Riverside
Riverside, CA
USA

TUTTLE, MARK
Oak Ridge National Laboratory
Oak Ridge, TN
USA

TWA, MICHAEL
The Ohio State University
Columbus, OH
USA

VATSAVAI, RANGA RAJU
Oak Ridge National Laboratory
Oak Ridge, TN
USA

VENETSANOPOULOS, A.N.
University of Toronto
Toronto, ONT
Canada

VIEUX, BAXTER E.
University of Oklahoma
Norman, OK
USA

VLACHOS, MICHAIL
IBM T.J. Watson Research Center
Hawthorne, NY
USA

VOISARD, AGNÈS
Fraunhofer ISST and FU Berlin
Berlin
Germany

VOSSELMAN, GEORGE
International Institute for Geo-Information Science and Earth Observation
Enschede
Netherlands

WALLER, LANCE A.
Emory University
Atlanta, GA
USA

WANG, X. SEAN
University of Vermont
Burlington, VT
USA

WEI, YAXING
George Mason University
Fairfax, VA
USA

WEIBEL, ROBERT
University of Zurich
Zurich
Switzerland

WEN, ZHEN
IBM T.J. Watson Research Center
Hawthorne, NY
USA

WIKLE, CHRISTOPHER K.
University of Missouri-Columbia
Columbia, MO
USA

WILFAHRT, RHETT
University of Minnesota
Minneapolis, MN
USA

WILSON, RONALD E.
National Institute of Justice
Washington, D.C.
USA

WOLFE, DEVON
MapInfo Canada
Toronto, ONT
Canada

WOLFSON, OURI
University of Illinois at Chicago
Chicago, IL
USA

WOLLE, THOMAS
NICTA
Eveleigh, NSW
Australia

WOLSHON, BRIAN
Louisiana State University
Baton Rouge, LA
USA

WOOD, JORDAN
University of Minnesota
Minneapolis, MN
USA

WOODSFORD, PETER
Laser-Scan
Cambridge
UK

WU, SHASHA
Spring Arbor University
Spring Arbor, MI
USA

XIA, YUNI
Purdue University
Indianapolis, IN
USA

XIE, QINGYUN (JEFFREY)
Oracle USA, Inc.
Nashua, NH
USA

XIONG, XIAOPENG
Purdue University
West Lafayette, IN
USA

YANG, CHAOWEI
George Mason University
Fairfax, VA
USA

YANG, HUI
San Francisco State University
San Francisco, CA
USA

YANG, JUE
University of North Texas
Denton, TX
USA

YANG, PHIL
George Mason University
Fairfax, VA
USA
NASA Geosciences Interoperability Office
Greenbelt, MD
USA

YANG, WENLI
George Mason University
Fairfax, VA
USA

YIU, MAN LUNG
Aalborg University
Aalborg
Denmark

YOUSSEF, MOUSTAFA
University of Maryland
College Park, MD
USA

YUAN, MAY
University of Oklahoma
Norman, OK
USA

ZAMBRE, RAJAN
Erallo Technologies Inc. Headquarters
Littleton, MA
USA

ZASLAVSKY, ILYA
University of California San Diego
San Diego, CA
USA

ZEINALIPOUR-YAZTI, DEMETRIOS
University of Cyprus
Nicosia
Cyprus

ZHANG, DONGHUI
Northeastern University
Boston, MA
USA

ZHANG, LIQIN
University of North Texas
Denton, TX
USA

ZHANG, PUSHENG
Microsoft Corporation
Redmond, WA
USA

ZHANG, RUI
The University of Melbourne
Parkville, VIC
Australia

ZHAO, PEISHENG
George Mason University
Fairfax, VA
USA

ZHOU, BIN
George Mason University
Fairfax, VA
USA
Tsinghua University
Beijing
China

ZHOU, NAIJUN
University of Maryland
College Park, MD
USA

ZHOU, XIAOBO
The Pennsylvania State University
University Park, PA
USA

ZHU, HAIJUN
George Mason University
Fairfax, VA
USA

ZIMÁNYI, ESTEBAN
The Free University of Brussels (ULB)
Brussels
Belgium

3-Value Indeterminacy

► Objects with Broad Boundaries

3D City Models

► Photogrammetric Applications

3D Models

► Photogrammetric Products

4-Intersection Calculus

► Mereotopology

4IM

► Dimensionally Extended Nine-Intersection Model (DE-9IM)

9-Intersection Calculus

► Mereotopology

9IM

► Dimensionally Extended Nine-Intersection Model (DE-9IM)

A* Algorithm

► Fastest-Path Computation

Absolute Positional Accuracy

► Positional Accuracy Improvement (PAI)

Abstract Features

► Feature Extraction, Abstract

Abstract Representation of Geographic Data

► Feature Catalogue

Abstraction

► Hierarchies and Level of Detail

Abstraction of GeoDatabases

MONIKA SESTER
Institute of Cartography and Geoinformatics,
Leibniz University of Hannover, Hannover, Germany

Synonyms

Model generalization; Conceptual generalization of databases; Cartographic generalization; Geographic data reduction; Multiple resolution database

Definition

Model generalization is used to derive a more simple and more easy to handle digital representation of geometric features [1]. It is being applied mainly by National Mapping Agencies to derive different levels of representations with less details of their topographic data sets, usually called Digital Landscape Models (DLM's). Model generalization is also called geodatabase abstraction, as it relates to generating a more simple digital representation of geometric objects in a database, leading to a considerable data reduction. The simplification refers to both the thematic diversity and the geometric complexity of the objects. Among the well known map generalization operations the following subset is used for model generalization: selection, (re-)classification, aggregation, and area collapse. Sometimes, also the reduction in the number of points to represent a geometric feature is applied in the model generalization process, although this is mostly considered a problem of cartographic generalization. This is achieved by line generalization operations.

Historical Background

Generalization is a process that has been applied by human cartographers to generate small scale maps from detailed ones. The process is composed of a number of elementary operations that have to be applied in accordance with each other in order to achieve optimal results. The difficulty is the correct interplay and sequencing of the operations, which depends on the target scale, the type of objects involved as well as constraints these objects are embedded in (e. g., topological constraints, geometric and semantic context, ...). Generalization is always subjective and requires the expertise of a human cartographer [2]. In the digital era, attempts to automate generalization have lead to the differentiation between model generalization and cartographic generalization, where the operations of model generalization are considered to be easier to automate than those of cartographic generalization.

After model generalization has been applied, the thematic and geometric granularity of the data set corresponds appropriately to the target scale. However, there might be some geometric conflicts remaining that are caused by applying signatures to the features as well as by imposing minimum distances between adjacent objects. These conflicts have to be solved by cartographic generalization procedures, among which typification and displacement are the most important (for a comprehensive overview see [3]). As opposed to cartographic generalization, model generalization processes have already achieved a high degree of automation. Fully automatic processes are available that are able to generalize large data sets, e. g., the whole of Germany ([4]).

Scientific Fundamentals

Operations of model generalization are selection, reclassification, aggregation, area collapse, and line simplification.

Selection

According to a given thematic and/or geometric property, objects are selected which are being preserved in the target scale. Typical selection criteria are object type, size or length. Objects fulfilling these criteria are preserved, whereas the others are discarded. In some cases, when an area partitioning of the whole data set has to be preserved, then the deleted objects have be replaced appropriately by neighboring objects.

Re-Classification

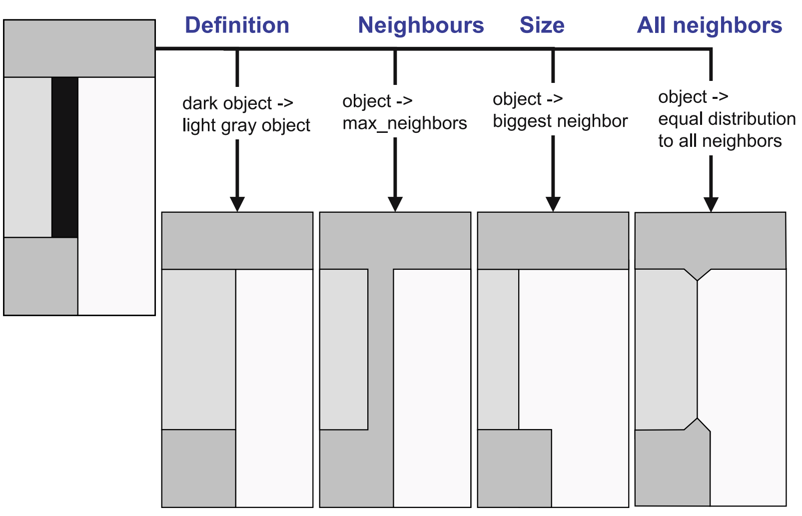
Often, the thematic granularity of the target scale is also reduced when reducing the geometric scale. This is realized by reclassification or new classification of object types. For example, in the German ATKIS system, when going from scale 1:25.000 to 1:50.000, the variation of settlement structures is reduced by merging two different settlement types to one class in the target scale.

Area Collapse

When going to smaller scales, higher dimensional objects may be reduced to lower dimensional ones. For instance, a city represented as an area is reduced to a point; an areal river is reduced to a linear river object. These reductions can be achieved using skeleton operations. For the area-to-line reduction, the use of the Medial Axis is popular, which is defined as the locus of points that have more than one closest neighbor on the polygon boundary. There are several approximations and special forms of axes (e. g., Straight Skeleton [5]). Depending on the object and the task at hand, there are forms that may be more favorable than others (e. g., [6,7]).

Aggregation

This is a very important operation that merges two or more objects into a single one, thus leading to a considerable amount of data reduction. Aggregation is often following a selection or area collapse process: when an object is too small (or unimportant) to be presented in the target scale, it has to be merged with a neighboring object. For the selection of the most appropriate neighbor, there are different strategies (see Fig. 1, e. g., selecting the neighbor according to thematic priority rules, the neighbor with the longest common boundary, the largest neighbor, or the area can be distributed equally to the neighbors ([7,8,9,10]). Another criterion is to select a neighbor which leads to a compact aggregated region and solve the whole problem as a global optimization process [11].



Abstraction of GeoDatabases, Figure 1
Different aggregation methods

Aggregation can also be performed when the objects are not topologically adjacent. Then, appropriate criteria for the determination of the neighborhood are needed as well as measures to fill the gaps between the neighboring polygons [12]. Aggregation can also be applied to other geometric features such as points and lines. This leads to point aggregations that can be approximated by convex hulls, or to aggregations of lines features.

Line Simplification

Line simplification is a very prominent generalization operation. Many operations have been proposed, mainly taking the relative distance between adjacent points and their relative context into account. The most well-known operator is the Douglas-Peucker-Algorithm [13].

Key Applications

The key application of database abstraction or model generalization is the derivation of less detailed data sets for different applications.

Cartographic Mapping

The production of small scale maps requires a detailed data set to be reduced in number and granularity of features. This reduction is achieved using database abstraction. It has to be followed by cartographic generalization procedures that are applied in order to generate the final symbolized map without graphical conflicts.

Visualization on Small Displays

The size of mobile display devices requires the presentation of a reduced number of features. To this end, the data can be reduced using data abstraction processes.

Internet Mapping – Streaming Generalization

Visualization of maps on the internet requires the transmission of an appropriate level of detail to the display of the remote user. To achieve an adequate data reduction that still ensures that the necessary information is communicated to the user, database abstraction methods are used. Also, it allows for the progressive transmission of more and more detailed information [14,15].

Spatial Data Analysis

Spatial analysis functions usually relate to a certain level of detail where the phenomena are best observed, e. g., for planning purposes, a scale of approximately 1:50.000 is very appropriate. Database abstraction can be used to generate this scale from base data sets. The advantage is that the level of detail is reduced while still preserving the geometric accuracy.

Future Directions

MRDB – Multiple Resolution Database

For topographic mapping, often data sets of different scales are provided by Mapping Agencies. In the past, these data sets were typically produced manually by generalization processes. With the availability of automatic generalization tools, such manual effort can be replaced. In order to make additional use of this lattice of data sets, the different scales are stored in a database where the individual objects in the different data sets are connected with explicit links. These links then allow for an efficient access of the corresponding objects in the neighboring scales, and thus an ease of movement up and down the different scales. There are several proposals for appropriate MRDB data structures, see e. g., [16]. The links can be created either in the generalization process or by matching existing data sets [17]. Although different approaches already exist, there is still research needed to fully exploit this data structure [18].

Data Update

An MRDB in principle offers the possibility of efficiently keeping the information in linked data sets up-to-date. The idea is to exploit the link structure and propagate the updated information to the adjacent and linked scales. There are several concepts for this, however, the challenge is to restrict the influence range to a manageable size [19].

Cross References

- ▶ Generalization, On-the-Fly
- ▶ Hierarchies and Level of Detail
- ▶ Map Generalization
- ▶ Mobile Usage and Adaptive Visualization
- ▶ Voronoi Diagram
- ▶ Web Mapping and Web Cartography

Recommended Reading

1. Grünreich, D.: Development of computer-assisted generalization on the basis of cartographic model theory. In: Müller J.C., Lagrange J.P., Weibel R. (eds.) GIS and Generalization – Methodology and Practice, pp. 47–55. Taylor & Francis, London (1995)
2. Spiess, E.: The need for generalization in a gis environment. In: Müller J.C., Lagrange J.P., Weibel R. (eds.) GIS and Generalization – Methodology and Practice, pp. 31–46. Taylor & Francis, London (1995)

3. Mackaness, W.A., Sarajakoski, L.T., Ruas, A.: Generalisation of Geographic Information: Cartographic Modelling and Applications. Published on behalf of the International Cartographic Association by Elsevier, Amsterdam (2007)
4. Urbanke, S., Dieckhoff, K.: The adv-project atkis generalization, part model generalization (in German). Kartographische Nachrichten **56**(4):191–196 (2006)
5. David, E., Erickson, J.: Raising roofs, crashing cycles, and playing pool: applications of a data structure for finding pairwise interactions. In: SCG 98: Proceedings of the 14th Annual Symposium on Computational Geometry. 58–67 (1998)
6. Chin, F.Y., Snoeyink, J., Wang, C.A.: Finding the medial axis of a simple polygon in linear time. In: Springer-Verlag (ed.) In ISAAC 95: Proceedings of the 6th International Symposium on Algorithms and Computation, 382–391, London, UK, 1995
7. Haunert, J.H., Sester, M.: Area collapse and road centerlines based on straight skeletons. Geoinformatica (2007, in press)
8. van Oosterom, P.: The gap-tree, an approach to 'on-the-fly' map generalization of an area partitioning. [20], 120–132
9. Podrenek, M.: Aufbau des DLM50 aus dem BasisDLM und Ableitung der DTK50 – Lösungsansatz in Niedersachsen. Kartographische Schriften Band 6, pp. 126–130. Kirschbaum Verlag, Bonn (2002)
10. van Smaalen, J.: Automated Aggregation of Geographic Objects. A new Approach to the Conceptual Generalisation of Geographic Databases. PhD thesis, Wageningen University, The Netherlands (2003)
11. Haunert, J.H., Wolff, A.: Generalization of land cover maps by mixed integer programming. In: Proceedings of 14th International Symposium on Advances in Geographic Information Systems, Arlington, Virginia, USA, 2006
12. Bundy, G., Jones, C., Furse, E.: Holistic generalization of large-scale cartographic data. [20], 106–119
13. Douglas, D., Peucker, T.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. The Canadian Cartographer **10**(2):112–122 (1973)
14. Brenner, C., Sester, M.: Continuous generalization for small mobile displays. In: Agouris, P., Croitoru, A. (eds.) Next Generation Geospatial Information, pp. 33–41, Taylor & Francis (2005)
15. Yang, B.: A multi-resolution model of vector map data for rapid transmission over the internet. Computers & Geosciences **31**(5): 569–578 (2005)
16. Balley, S., Parent, C., Spaccapietra, S.: Modelling geographic data with multiple representation. International Journal of Geographical Information Science **18**(4):327–352 (2004)
17. Hampe, M., Sester, M., Harrie, L.: Multiple representation databases to support visualisation on mobile devices. In: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 35, ISPRS, Istanbul (2004)
18. Sheeren, D., Mustière, S., Zucker, J.D.: Consistency assessment between multiple representations of geographical databases: a specification-based approach. In: Proc. of the 11th International Symposium on Spatial Data Handling, Leicester, UK, 2004
19. Haunert, J.H., Sester, M.: Propagating updates between linked datasets of different scales. In: Proceedings of 22nd International Cartographic Conference, 9–16, La Coruna, Spain, Juli 2005
20. Müller, J.C., Lagrange, J.P., Weibel, R. (eds.): GIS and Generalization – Methodology and Practice. In: Müller J.C., Lagrange J.P., Weibel R. (eds.) GIS and Generalization – Methodology and Practice, Taylor & Francis, London (1995)

Access Control

▶ Privacy Threats in Location-Based Services

Access Method, High-Dimensional

▶ Indexing, X-Tree

Access Structures for Spatial Constraint Databases

▶ Indexing Spatial Constraint Databases

Accuracy

▶ Uncertain Environmental Variables in GIS

Accuracy, Map

▶ Imprecision and Spatial Uncertainty

Accuracy, Spatial

▶ Imprecision and Spatial Uncertainty

Active Data Mining

▶ Gaussian Process Models in Spatial Data Mining

ActiveX Components

▶ MapWindow GIS

Activities and Occurrences

▶ Processes and Events

Activities, Flexible

▶ Time Geography

Activities, Fixed

▶ Time Geography

Activity

▶ Temporal GIS and Applications

Activity Analysis

▶ Time Geography

Activity Theory

▶ Time Geography

Acyclic Directed Graph

▶ Hierarchies and Level of Detail

Ad-Hoc Localization

▶ Localization, Cooperative

Adaptation

▶ Geospatial Semantic Web: Personalisation

Adaptation, Complete

▶ User Interfaces and Adaptive Maps

Adaption, Complete

▶ Mobile Usage and Adaptive Visualization

Adaptive

▶ User Interfaces and Adaptive Maps

Adaptive, Context-Aware

▶ Mobile Usage and Adaptive Visualization

Adaptive Visualization

▶ Mobile Usage and Adaptive Visualization

Aerial

▶ Photogrammetric Applications

Aerial Imagery

▶ Photogrammetric Sensors

Affordance

▶ Wayfinding: Affordances and Agent Simulation

Agent-Based Models

▶ Geographic Dynamics, Visualization and Modeling

Agent Simulation

▶ Wayfinding: Affordances and Agent Simulation

Aggregate Queries, Progressive Approximate

IOSIF LAZARIDIS, SHARAD MEHROTRA
Department of Computer Science,
University of California, Irvine, CA, USA

Synonyms

Approximate aggregate query; On-line aggregation

Definition

Aggregate queries generally take a set of objects as input and produce a single scalar value as output, summarizing one aspect of the set. Commonly used aggregate types include MIN, MAX, AVG, SUM, and COUNT.

If the input set is very large, it might not be feasible to compute the aggregate precisely and in reasonable time. Alternatively, the precise value of the aggregate may not even be needed by the application submitting the query, e. g., if the aggregate value is to be mapped to an 8-bit color code for visualization. Hence, this motivates the use of *approximate* aggregate queries, which return a value close to the exact one, but at a fraction of the time.

Progressive approximate aggregate queries go one step further. They do not produce a single approximate answer, but continuously refine the answer as time goes on, progressively improving its quality. Thus, if the user has a fixed deadline, he can obtain the best answer within the allotted time; conversely, if he has a fixed answer accuracy requirement, the system will use the least amount of time to produce an answer of sufficient accuracy. Thus, progressive approximate aggregate queries are a flexible way of implementing aggregate query answering.

Multi-Resolution Aggregate trees (MRA-trees) are spatial – or in general multi-dimensional – indexing data structures, whose nodes are augmented with aggregate values for all the indexed subsets of data. They can be used very efficiently to provide an implementation of progressive approximate query answering.

Historical Background

Aggregate queries are extremely useful because they can summarize a huge amount of data by a single number. For example, many users expect to know the average and highest temperature in their city and are not really interested in the temperature recorded by all environmental monitoring stations used to produce this number. The simplest aggregate query specifies a selection condition specifying the subset of interest, e. g., "all monitoring stations in Irvine" and an aggregate type to be computed, e. g., "MAX temperature".

The normal way to evaluate an aggregate query is to collect all data in the subset of interest and evaluate the aggregate query over them. This approach has two problems: first, the user may not need to know that the temperature is 34.12°C, but 34 ± 0.5°C will suffice; second, the dataset may be so large that exhaustive computation may be infeasible. These observations motivated researchers to devise approximate aggregate query answering mechanisms.

Off-line *synopsis* based strategies, such as histograms [5], samples [1], and wavelets [2] have been proposed for approximate query processing. These use small data summaries that can be processed very easily to answer a query at a small cost. Unfortunately, summaries are inherently unable to adapt to the query requirements. The user usually has no way of knowing how good an approximate answer is and, even if he does, it may not suffice for his goals. Early synopsis based techniques did not provide any guarantees about the quality of the answer, although this has been incorporated more recently [3].

Online aggregation [4] was proposed to deal with this problem. In online aggregation, the input set is sampled continuously, a process which can, in principle, continue until this set is exhausted, thus providing an answer of arbitrarily good quality; the goal is, however, to use a sample of small size, thus saving on performance while giving a "good enough" answer. In online aggregation, a running aggregate is updated progressively, finally converging to the exact answer if the input is exhausted. The sampling usually occurs by sampling either the entire data table or a subset of interest one tuple at a time; this may be expensive, depending on the size of the table, and also its organization: if tuples are physically ordered in some way, then sampling may need to be performed with random disk accesses, which are costiercompared to sequential accesses.

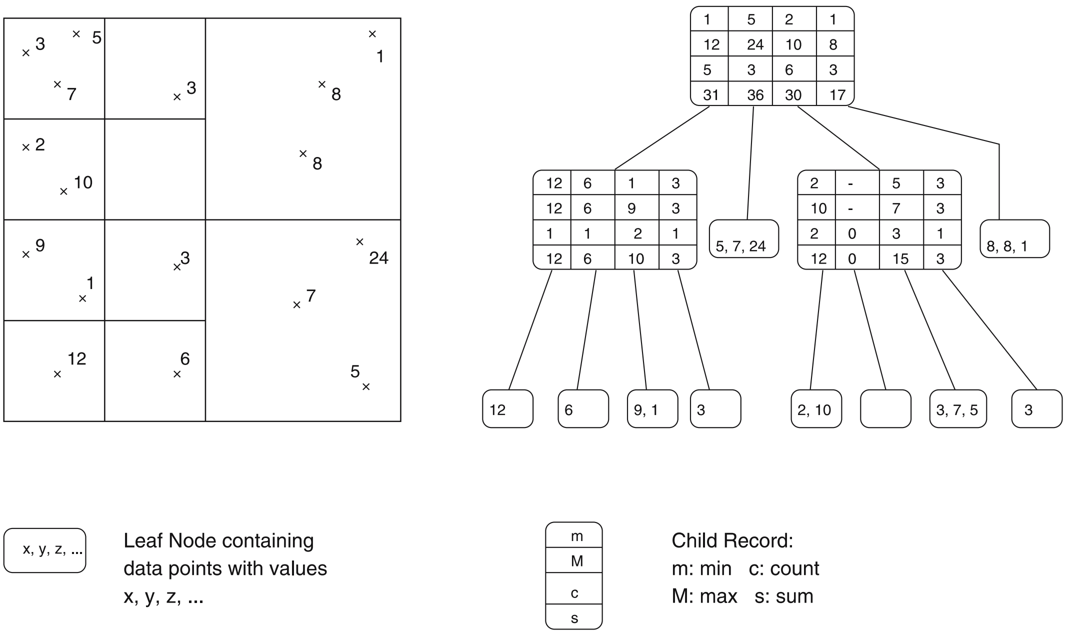
Multi-resolution trees [7] were designed to deal with the limitations of established synopsis-based techniques and sampling-based online aggregation. Unlike off-line synopses, MRA-trees are flexible and can adapt to the characteristics of the user's quality/time requirements. Their advantage over sampling is that they help queries quickly zero in on the subset of interest without having to process a great number of tuples individually. Moreover, MRA-trees provide deterministic answer quality guarantees to the user that are easy for him to prescribe (when he poses his query) and to interpret (when he receives the results).

Scientific Fundamentals

Multi-dimensional index trees such as R-trees, quad-trees, etc., are used to index data existing in a multi-dimensional domain. Consider a d-dimensional space $\mathbb{R}^d$ and a finite set of points (input relation) $S \subset \mathbb{R}^d$. Typically, for spatial applications, $d \in \{2, 3\}$. The aggregate query is defined as a pair (agg, R^Q) where agg is an aggregate function (e. g., MIN, MAX, SUM, AVG, COUNT) and $R^Q \subset \mathbb{R}^d$ is the query region. The query asks for the evaluation of agg over all tuples in S that are in region R^Q. Multi-dimensional index trees organize this data via a hierarchical decomposition of the space $\mathbb{R}^d$ or grouping of the data in S. In either case, each node N indexes a set of data tuples contained in its subtree which are guaranteed to have values within the node's region R^N.

MRA-trees [7] are generic data techniques that can be applied over any standard multi-dimensional index method; they are not yet another indexing technique. They modify the underlying index by adding the value of the agg over all data tuples indexed by (i. e., in the sub-tree of) N to each tree node N. Only a single such value, e. g., MIN, may be stored, but in general, all aggregate types can be used without much loss of performance. An example of an MRA-quad-tree is seen in Fig. 1.

The key observation behind the use of MRA-trees is that the aggregate value of all the tuples indexed by a node N is known by just visiting N. Thus, in addition to the performance benefit of a standard spatial index (visiting only a fraction of selected tuples, rather than the entire



Aggregate Queries, Progressive Approximate, Figure 1 Example of an MRA-quad-tree

set), the MRA-tree also avoids traversing the entire subtree of nodes contained within the query region. Nodes that partially overlap the region may or may not contribute to the aggregate, depending on the spatial distribution of points within them. Such nodes can be further explored to improve performance. This situation is seen in Fig. 2: nodes at the perimeter of the query (set N_p) can be further explored, whereas nodes at the interior (N_c) need not be.

The progressive approximation algorithm (Fig. 3) has three major components:

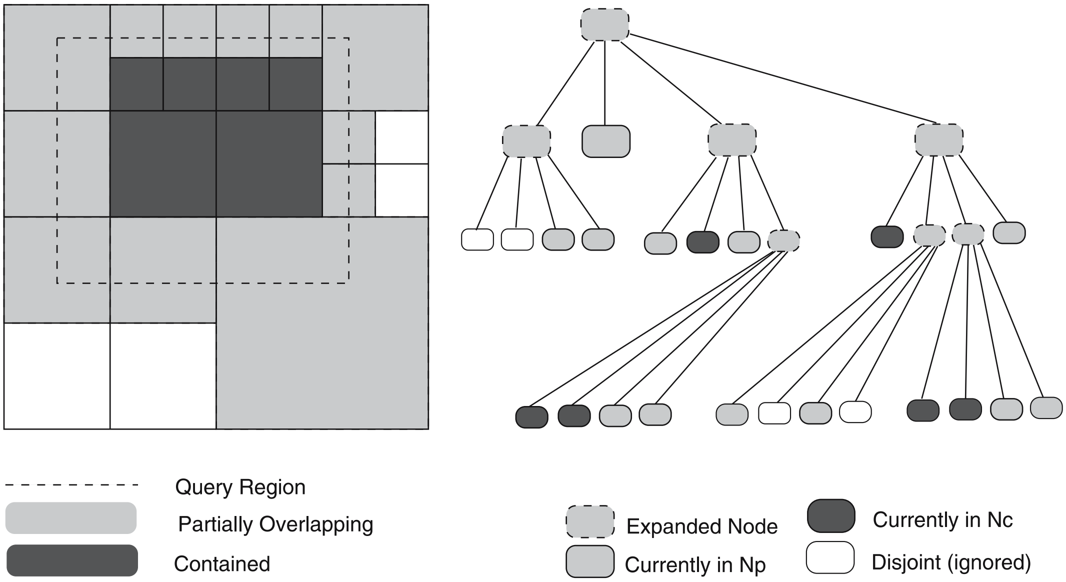
- Computation of a deterministic *interval of confidence* guaranteed to contain the aggregate value, e. g., [30, 40].
- *Estimation* of the aggregate value, e. g., 36.2.
- A *traversal policy* which determines which node to explore next by visiting its children nodes.

The interval of confidence can be calculated by taking the set of nodes partially overlapping/contained in the query into account (Fig. 2). The details of this for all the aggregate types can be found in [7]. For example, if the SUM of all contained nodes is 50 and the SUM of all partially overlapping nodes is 15, then the interval is [50, 65] since all the tuples in the overlapping nodes could either be outside or inside the query region.

There is no single best way for aggregate value estimation. For example, taking the middle of the interval has the advantage of minimizing the worst-case error. On the other hand, intuitively, if a node barely overlaps with the query, then it is expected that its overall contribution to the query will be slight. Thus, if in the previous example there are two partially overlapping nodes, A and B, with SUM(A) = 5 and SUM(B) = 15, and 30% of A and 50% of B overlaps with the query respectively, then a good estimate of the SUM aggregate will be $50 + 5 \times 0.3 + 15 \times 0.5 = 59$.

Finally, the traversal policy should aim to shrink the interval of confidence by the greatest amount, thus improving the accuracy of the answer as fast as possible. This is achieved by organizing the partially overlapping nodes using a priority queue. The queue is initialized with the root node and subsequently the front node of the queue is repeatedly picked, its children examined, the confidence interval and aggregate estimate is updated, and the partially overlapping children are placed in the queue. Our example may show the preference to explore node B before A since it contributed more (15) to the uncertainty inherent in the interval of confidence than B (5). Detailed descriptions of the priority used for the different aggregate types can be found in [7].

Performance of MRA-trees depends on both the underlying data structure used as well as the aggregate type and query selectivity. MIN and MAX queries are typically evaluated very efficiently since the query processing system uses the node aggregates to quickly zero in on a few candidate nodes that contain the minimum value; very rarely is the entire perimeter needed to compute even the exact answer. Query selectivity affects processing speed; like all multi-dimensional indexes, performance

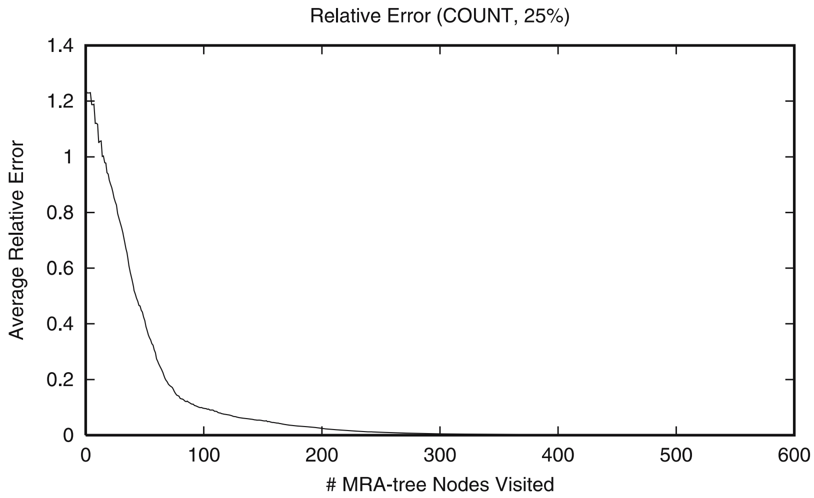


Aggregate Queries, Progressive Approximate, Figure 2 A snapshot of MRA-tree traversal

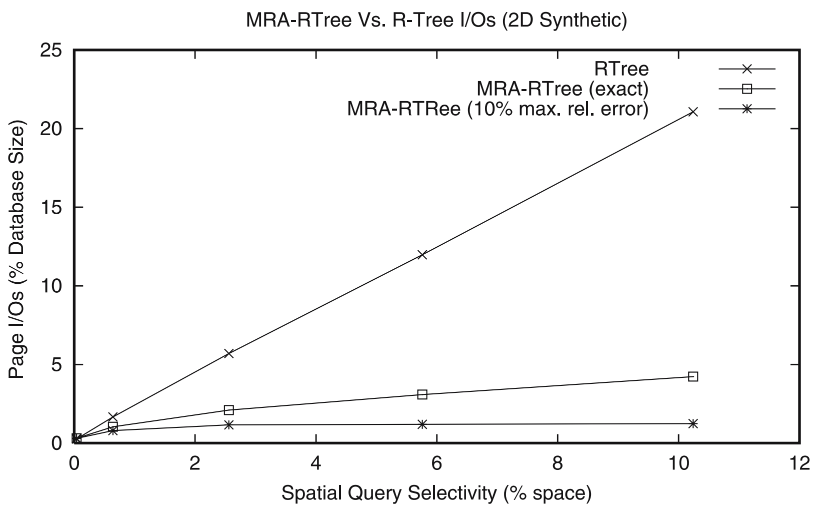
```
class Query
{
    R^Q: Region;
    agg: enum {MIN, MAX, COUNT, SUM, AVG};
    N_c:=∅,N_p:=∅: set of Node;
};

Query::Query(queryR: Region, root: Node, aggType)
{
   R^Q := queryR;
   agg := aggType;
   if (R^Q ∩ R^root = R^root) {
      N_c := {root};
   else
      N_p := {root};
};

Query::nextAnswer(var ans: Real, var low: Real, var high: Real)
{
   if (N_p = ∅) {
      estimate ans from N_c;
      low := high := ans;
   }
   else {
      remove node N from N_p;
      forall children N_i : i = 1, . . . , n_N of node N {
         if (R^Q ∩ R^N = ∅)
            ignore;
         else if (R^Q ∩ R^N = R^N)
            insert N_i into N_c;
         else
            if N_i is useful
               insert N_i into N_p;
      }
      estimate ans from N_p, N_c;
      calculate low, high from N_p, N_c;
   }
};
```

Aggregate Queries, Progressive Approximate, Figure 3 Progressive approximation algorithm

A



Aggregate Queries, Progressive Approximate, Figure 4 Answer error improves as more MRA-tree nodes are visited



Aggregate Queries, Progressive Approximate, Figure 5 MRA-R-tree performance compared to regular R-tree

degrades as a higher fraction of the input table *S* is selected. However, unlike traditional indexes, the degradation is more gradual since the "interior" area of the query region is not explored. A typical profile of answer error as a function of the number of nodes visited can be seen in Fig. 4.

MRA-trees use extra space (to store the aggregates) in exchange for time. If the underlying data structure is an R-tree, then storage of aggregates in tree nodes results in decreased fanout since fewer bounding rectangles and their accompanying aggregate values can be stored within a disk page. Decreased fanout may imply increased height of the tree. Fortunately, the overhead of aggregate storage does not negatively affect performance since it is counterbalanced by the benefits of partial tree exploration. Thus, even for computing the exact answer, MRA-trees are usually faster than regular R-trees and the difference grows even if a small error, e. g., in the order of 10%, is allowed (Fig. 5).

Key Applications

Progressive approximate aggregate queries using a multiresolution tree structure can be used in many application domains when data is either large, difficult to process, or the exact answer is not needed.

On-line Analytical Processing: On-line analytical processing (OLAP) is often applied to huge transaction datasets, such as those produced by merchants or other geographically distributed enterprises. If these data are indexed using an MRA-tree, such as aggregate queries, the most frequent type of query found in OLAP can be processed efficiently.

Wireless Sensor Networks: Sensor networks consist of numerous small sensors deployed over a geographical region of interest. Interestingly, sensors are often organized in a routing tree leading to an access point from which data is forwarded to the data infrastructure. This routing tree itself could become a spatial index, thus limiting the number of hops of wireless communication needed to obtain the aggregate value. Thus, fewer hops of wireless communication analogous to disk I/Os in disk-based data structures such as R-trees, will be necessary.

Virtual Reality and Visualization: Information about a geographical region is often presented in visual form, in either a static or a dynamic presentation (e. g., a virtual fly-through). Queries may come at a very high rate (equal to the frame rate), whereas the precision of the visualization is inherently limited by the color coding and limitations of human perception. Approximate aggregate queries are thus ideally suited to drive interactive visualizations [11].

Future Directions

A limitation of MRA-trees is that they have to maintain the aggregate values at each node of the tree. Thus, whenever a data insertion and deletion takes place, all nodes in the path from the root to the modified leaf have to be updated. This cost may be significant, e. g., in applications with frequent updates, such as those involving moving objects. This extra cost may be reduced if updates are deferred; this would improve performance, but with an accompanying loss of accuracy.

Beyond aggregate queries, progressive approximation can also be used in queries producing a set of objects as output. Unlike aggregate queries that admit to a natural definition of accuracy (the length of the confidence interval), there is no clear metric to assess the quality of set-based answers. Precision and recall used in information retrieval systems may quantify the purity and completeness of the answer set [8], but more elaborate methods can be devised, particularly if the answer set is visualized in a GIS system. While datasets continue to exponentially grow in size, visualization media and the human perceptual system does not, and hence, it will be useful to adapt query processing to their limitations rather than to process data exhaustively at a great cost, but with no observable benefit for the user.

Cross References

- ► Multi-Resolution Aggregate Tree
- ► Progressive Approximate Aggregation

Recommended Reading

1. Acharya, S., Gibbons, P., Poosala, V., Ramaswamy, S.: Joint synopses for approximate query answering. SIGMOD '99: Proceedings of the 1999 ACM SIGMOD international conference on Management of data, pp. 275–286. ACM Press, New York, NY, USA (1999)
2. Chakrabarti, K., Garofalakis, M.N., Rastogi, R., Shim, K.: Approximate query processing using wavelets. In: VLDB '00: Proceedings of the 26th International Conference on Very Large Data Bases, pp. 111–122. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2000)
3. Garofalakis, M., Kumar, A.: Wavelet synopses for general error metrics. ACM Trans. Database Syst. **30**(4), 888–928 (2005)
4. Hellerstein, J.M., Haas, P.J., Wang, H.J.: Online aggregation. In: SIGMOD '97: Proceedings of the 1997 ACM SIGMOD International conference on Management of data, pp. 171–182. ACM Press, New York, NY, USA (1997)
5. Ioannidis, Y.E., Poosala, V.: Histogram-based approximation of set-valued query-answers. In: VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases, pp. 174–185. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1999)
6. Karras, P., Mamoulis, N.: One-pass wavelet synopses for maximum-error metrics. In: VLDB '05: Proceedings of the 31st international conference on Very large data bases, pp. 421–432, Trondheim, Norway. VLDB Endowment (2005)
7. Lazaridis, I., Mehrotra, S.: Progressive approximate aggregate queries with a multi-resolution tree structure. In: SIGMOD '01: Proceedings of the 2001 ACM SIGMOD international conference on Management of data, pp. 401–412. ACM Press, New York, NY, USA (2001)
8. Lazaridis, I., Mehrotra, S.: Approximate selection queries over imprecise data. In: ICDE '04: Proceedings of the 20th International Conference on Data Engineering, p.140. Washington, DC, USA, IEEE Computer Society (2004)
9. Lenz, H.J., Jurgens, M.: The Ra*-tree: An improved r-tree with materialized data for supporting range queries on olap-data. In: DEXA Workshop, 1998
10. Papadias, D., Kalnis, P., Zhang, J., Tao, Y.: Efficient OLAP operations in spatial data warehouses. In: SSTD '01: Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases, pp. 443–459. London, UK, Springer-Verlag (2001)
11. Porkaew, K., Lazaridis, I., Mehrotra, S., Winkler, R.:Database Support for Situational Awareness. In: Vassiliop, M.S., Huang, T.S. Computer-Science Handbook for Displays – Summary of Findings from the Army Research Lab's Advanced Displays & Interactive Displays Federated Laboratory. Rockwell Scientific Company (2001)

Aggregation

► Hierarchies and Level of Detail

Aggregation Query, Spatial

DONGHUI ZHANG
College of Computer and Information Science,
Northeastern University, Boston, MA, USA

Synonyms

Spatial aggregate computation

Definition

*Given a set O of weighted point objects and a rectangular query region r in the d-dimensional space, the **spatial aggregation query** asks the total weight of all objects in O which are contained in r.*

This query corresponds to the SUM aggregation. The COUNT aggregation, which asks for the number of objects in the query region, is a special case when every object has equal weight.

The problem can actually be reduced to a special case, called the ***dominance-sum query***. An object o_1 dominates another object o_2 if o_1 has larger value in all dimensions. The dominance-sum query asks for the total weight of objects dominated by a given point p. It is a special case of the spatial aggregation query, when the query region is described by two extreme points: the lower-left corner of space, and p.

The spatial aggregation query can be reduced to the dominance-sum query in the 2D space, as illustrated below. Given a query region r (a 2D rectangle), let the four corners of r be *lowerleft*, *upperleft*, *lowerright*, and *upperright*. It is not hard to verify that the spatial aggregate regarding to r is equal to

$$\begin{aligned} & dominancesum(upperright) \\ - \; & dominancesum(lowerright) \\ - \; & dominancesum(upperleft) \\ + \; & dominancesum(lowerleft) \end{aligned}$$

Historical Background

In computational geometry, to answer the dominance-sum query, an *in-memory* and *static* data structure called the **ECDF-tree** [2] can be used. The ECDF-tree is a *multi-level data structure*, where each level corresponds to a different dimension. At the first level (also called *main branch*), the d-dimensional ECDF-tree is a full binary search tree whose leaves store the data points, ordered by their position in the first dimension. Each internal node of this binary search tree stores a *border* for all the points in the left sub-tree. The *border* is itself a (d-1)-dimensional ECDF-tree; here points are ordered by their positions in the second dimension. The collection of all these border trees forms the second level of the structure. Their respective borders are (d-2)-dimensional ECDF-trees (using the third dimension and so on). To answer a dominance-sum query for point $p = (p_1, \ldots, p_d)$, the search starts with the root of the first level ECDF-tree. If p_1 is in the left sub-tree, the search continues recursively on the left sub-tree. Otherwise, two queries are performed, one on the right sub-tree and the other on the border; the respective results are then added together.

In the fields of GIS and spatial databases, one seeks for *disk-based* and *dynamically updateable* index structures. An approach is to externalize and dynamize the ECDE-tree. To *dynamize* a static data structure some standard techniques can be used [4]. For example, the *global rebuilding* [8] or the *logarithmic method* [3]. To *externalize* an internal-memory data structure, a widely used method is to augment it with block-access capabilities [11]. Unfortunately, this approach is either very expensive in query cost, or very expensive in index size and update cost.

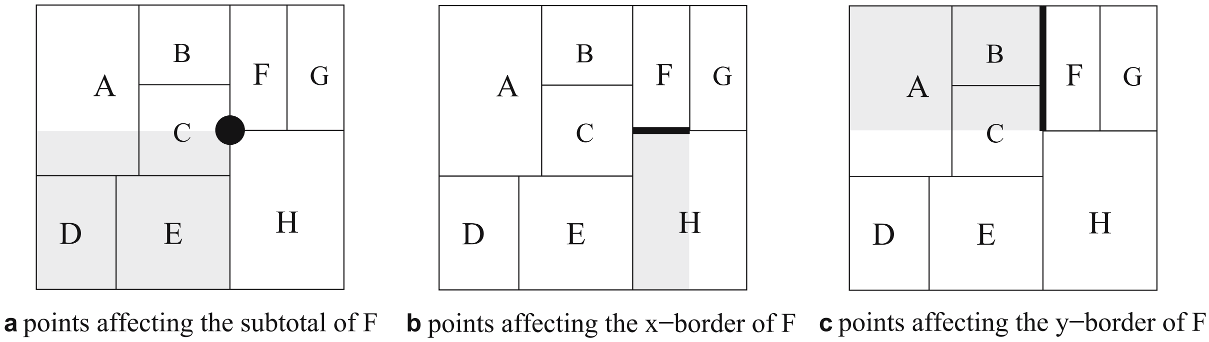
Another approach to solve the spatial aggregation query is to index the data objects with a multi-dimensional access method like the *R*-tree* [1]. The R*-tree (and the other variations of the R-tree) clusters nearby objects into the same disk page. An index entry is used to reference each disk page. Each index entry stores the *minimum bounding rectangle (MBR)* of objects in the corresponding disk page. The index entries are then recursively clustered based on proximity as well. Such multi-dimensional access methods provide efficient *range* query performance in that subtrees whose MBRs do not intersect the query region can be pruned. The spatial aggregation query can be reduce to the range search: retrieve the objects in the query region and aggregate their weights on-the-fly. Unfortunately, when the query region is large, the query performance is poor.

An optimization proposed by [7,9] is to store, along with each index entry, the total weight of objects in the referenced sub-tree. The index is called the aggregate R-tree, or *aR-tree* in short. Such aggregate information can improve the aggregation query performance is that if the query region fully contains the MBR of some index entry, the total weight stored along with the index entry contributes to the answer, while the sub-tree itself does not need to be examined. However, even with this optimization the query effort is still affected by the size of the query region.

Scientific Fundamentals

This section presents a better index for the dominance-sum query (and in turn the spatial aggregation query) called the *Box-Aggregation Tree*, or ***BA-tree*** in short.

The BA-tree is an augmented k-d-B-tree [10]. The k-d-B-tree is a disk-based index structure for multi-dimensional point objects. Unlike the R-tree, the k-d-B-tree indexes the whole space. Initially, when there are only a few objects, the k-d-B-tree uses a single disk page to store them. The page is responsible for the whole space in the sense that any new object, wherever it is located in space, should be inserted to this page. When the page overflows, it is split into two using a hyperplane corresponding to a single dimension. For instance, order all objects based on



Aggregation Query, Spatial, Figure 1 The BA-tree is a k-d-B-tree with augmented border information

dimension one, and move the half of the objects with larger dimension-one values to a new page. Each of these two disk pages are referenced by an index entry, which contains a *box*: the space the page is responsible for. The two index entries are stored in a newly created index page. As more split happens, the index page contains more index entries.
For ease of understanding let's focus discussion on the 2D space. Figure 1 shows an exemplary index page of a BA-tree in the 2D space. As in the k-d-B-tree, each index record is associated with a *box* and a *child* pointer. The boxes of records in a page do not intersect and their union creates the box of the page.
As done in the ECDE-tree, each index record in the k-d-B-tree can be augmented with some *border* information. The goal is that a dominance-sum query can be answered by following a single sub-tree (in the main branch). Suppose in Fig. 1a, there is a query point contained in the box of record *F*. The points that may affect the dominance-sum query of a query point in *F.box* are those dominated by the upper-right point of *F.box*. Such points belong in four groups: (1) the points contained in *F.box*; (2) the points dominated by the low point of *F* (in the shadowed region of Fig. 1a); (3) the points below the lower edge of *F.box* (Fig. 1b); and (4) the points to the left of the left edge of *F.box* (Fig. 1c).
To compute the dominance-sum for points in the first group, a recursive traversal of subtree(*F*) is performed. For points in the second group, in record *F* a single value (called *subtotal*) is kept, which is the total value of all these points. For computing the dominance-sum in the third group, an *x-border* is kept in *F* which contains the *x* positions and values of all these points. This dominance-sum is then reduced to a 1D dominance-sum query for the border. It is then sufficient to maintain these *x* positions in a 1D BA-tree. Similarly, for the points in the fourth group, a *y-border* is kept which is a 1D BA-tree for the *y* positions of the group's points.
To summarize, the 2D BA-tree is a k-d-B-tree where each index record is augmented with a single value *subtotal* and two 1D BA-trees called *x-border* and *y-border*, respectively. The computation for a dominance-sum query at point *p* starts at the root page *R*. If *R* is an index node, it locates the record *r* in *R* whose box contains *p*. A 1D dominance-sum query is performed on the *x*-border of *r* regarding *p.x*. A 1D dominance-sum query is performed on the *y*-border of *r* regarding *p.y*. A 2D dominance-sum query is performed recursively on page(*r.child*). The final query result is the sum of these three query results plus *r.subtotal*.
The insertion of a point *p* with value *v* starts at the root *R*. For each record *r* where *r.lowpoint* dominates *p*, *v* is added to *r.subtotal*. For each *r* where *p* is below the *x*-border of *r*, position *p.x* and value *v* are added to the *x*-border. For each record *r* where *p* is to the left of the *y*-border of *r*, position *p.y* and value *v* are added to the *y*-border. Finally, for the record *r* whose box contains *p*, *p* and *v* are inserted in the subtree(*r.child*). When the insertion reaches a leaf page *L*, a leaf record that contains point *p* and value *v* is stored in *L*.
Since the BA-tree aims at storing only the aggregate information, not the objects themselves, there are chances where the points inserted are not actually stored in the index, thus saving storage space. For instance, if a point to be inserted falls on some border of an index record, there is no need to insert the point into the sub-tree at all. Instead, it is simply kept in the border that it falls on. If the point to be inserted falls on the low point of an internal node, there is even no need to insert it in the border; rather, the *subtotal* value of the record is updated.
The BA-tree extends to higher dimensions in a straightforward manner: a *d*-dimensional BA-tree is a k-d-B-tree where each index record is augmented with one *subtotal* value and *d* borders, each of which is a (*d*-1)-dimensional BA-tree.

Key Applications

One key application of efficient algorithms for the spatial aggregation query is interactive GIS systems. Imagine a user interacting with such a system. She sees a map on the computer screen. Using the mouse, she can select a rectangular region on the map. The screen zooms in to the selected region. Besides, some statistics about the selected region, e. g. the total number of hotels, total number of residents, and so on, can be quickly computed and displayed on the side.

Another key application is in data mining, in particular, to compute *range-sums* over data cubes. Given a d-dimensional array A and a query range q, the range-sum query asks for the total value of all cells of A in range q. It is a crucial query for online analytical processing (OLAP). The best known in-memory solutions for data cube range-sum appear in [5,6]. When applied to this problem, the BA-tree differs from [6] in two ways. First, it is disk-based, while [6] presents a main-memory structure. Second, the BA-tree partitions the space based on the data distribution while [6] does partitioning based on a uniform grid.

Future Directions

The update algorithm for the BA-tree is omitted from here, but can be found in [12]. Also discussed in [12] are more general queries, such as spatial aggregation over objects with extent.

The BA-tree assumes that the query region is an axis-parallel box. One practical direction of extending the solution is to handle arbitrary query regions, in particular, polygonal query regions.

Cross References

▸ Aggregate Queries, Progressive Approximate
▸ OLAP, Spatial

References

1. Beckmann, N., Kriegel, H.P., Schneider, R., Seeger, B.: The R*-tree: An Efficient and Robust Access Method for Points and Rectangles. In: SIGMOD, pp. 322–331 (1990)
2. Bentley, J.L.: Multidimensional Divide-and-Conquer. Commun.s ACM **23**(4), 214–229 (1980)
3. Bentley, J.L., Saxe, N.B.: Decomposable Searching Problems I: Static-to-Dynamic Transformations. J. Algorithms **1**(4), 301–358 (1980)
4. Chiang, Y., Tamassia R.: Dynamic Algorithms in Computational Geometry. Proc. of the IEEE, Special Issue on Computational Geometry **80**(9), 1412–1434 (1992)
5. Chung, C., Chun, S., Lee, J., Lee, S.: Dynamic Update Cube for Range-Sum Queries. In: VLDB, pp. 521–530 (2001)
6. Geffner, S., Agrawal, D., El Abbadi, A.: The Dynamic Data Cube. In: EDBT, pp: 237–253 (2000)
7. Lazaridis, I., Mehrotra, S.: Progressive Approximate Aggregate Queries with a Multi-Resolution Tree Structure. In: SIGMOD, pp. 401–412 (2001)
8. Overmars, M.H.: The Design of Dynamic Data Structures. Lecture Notes in Computer Science, vol. 156, Springer, Heidelberg (1983)
9. Papadias, D., Kalnis, P., Zhang, J., Tao, Y.: Efficient OLAP Operations in Spatial Data Warehouses. In: Jensen, C.S.; Schneider, M.; Seeger, B.; Tsotras, V.J. (eds.) Advances in Spatial and Temporal Databases, 7th International Symposium, SSTD 2001, Redondo Beach, July 2001. Lecture Notes in Computer Science, vol. 2121, pp. 443–459. Springer, Heidelberg (2001)
10. Robinson, J.: The K-D-B Tree: A Search Structure For Large Multidimensional Dynamic Indexes. In: SIGMOD, pp 10–18 (1981)
11. Vitter, J.S.: External Memory Algorithms and Data Structures. ACM Comput. Surveys **33**(2), 209–271 (2001)
12. Zhang, D., Tsotras, V.J., Gunopulos, D.: Efficient Aggregation over Objects with Extent. In: PODS, pp. 121–132 (2002)

Air Borne Sensors

▸ Photogrammetric Sensors

akNN

▸ Nearest Neighbors Problem

Algorithm

▸ Data Structure

All-K-Nearest Neighbors

▸ Nearest Neighbors Problem

All-Lanes-Out

▸ Contraflow for Evacuation Traffic Management

All-Nearest-Neighbors

▸ Nearest Neighbors Problem

Ambient Spatial Intelligence

▸ Geosensor Networks, Qualitative Monitoring of Dynamic Fields

Ambiguity

▸ Uncertainty, Semantic
▸ Retrieval Algorithms, Spatial

Analysis, Robustness

▶ Multicriteria Decision Making, Spatial

Analysis, Sensitivity

▶ Multicriteria Decision Making, Spatial

Anamolies

▶ Data Analysis, Spatial

Anchor Points

▶ Wayfinding, Landmarks

Anchors, Space-Time

▶ Time Geography

Anomaly Detection

▶ Homeland Security and Spatial Data Mining
▶ Outlier Detection

Anonymity

▶ Cloaking Algorithms for Location Privacy

Anonymity in Location-Based Services

▶ Privacy Threats in Location-Based Services

Anonymization of GPS Traces

▶ Privacy Preservation of GPS Traces

ANSI NCITS 320-1998

▶ Spatial Data Transfer Standard (SDTS)

Application

▶ Photogrammetric Applications

Application Schema

JEAN BRODEUR[1], THIERRY BADARD[2]
[1] Center for Topographic Information, Natural Resources Canada, Sherbrooke, QC, Canada
[2] Center for Research in Geomatics (CRG), Department of Geomatic Science, Université Laval, Québec, Canada

Synonyms

ISO/TC 211; Conceptual model; Conceptual schema; Data schema; Data models; Object schema; Object model

Definition

In the context of geographic information and ISO/TC 211 vocabulary, an application schema consists in an application level conceptual schema rendering to a certain level of detail a universe of discourse described as data. Such data is typically required by one or more applications [1]. Typically, additional information not found in the schema is included in a feature catalogue to semantically enrich the schema. Levels of details regarding schemata (models) and catalogues (data dictionaries) are described in the cross-references.

Main Text

An application schema documents the content and the logical structure of geographic data along with manipulating and processing operations of the application to a level of details that allows developers to set up consistent, maintainable, and unambiguous geographic databases and related applications [2]. As such, an application schema contributes to both the semantics of geographic data and describes the structure of the geographic information in a computer-readable form. It specifies spatial and temporal objects, and may also specify reference systems and data quality elements used to depict geographic features. It also supports the use of the geographic data appropriately (i. e. fitness for use). Typically, an application schema is depicted in a formal conceptual schema language.

Cross References

▶ Modeling with ISO 191xx Standards
▶ Modeling with Pictogrammic Languages

Recommended Reading

1. ISO/TC211 ISO19109:2005 Geographic information – Rules for application schema (ISO) (2005)
2. Brodeur, J. Bédard, Y. and Proulx, M. J.: Modelling Geospatial Application Databases using UML-based Repositories Aligned

with International Standards in Geomatics. Eighth ACM Symposium on Advances in Geographic Information Systems (ACMGIS) (2000)

Approximate Aggregate Query

► Aggregate Queries, Progressive Approximate

Approximation

THOMAS BITTNER[1], JOHN G. STELL[2]
[1] Department of Philosophy; Department of Geography, State University of New York at Buffalo, Buffalo, NY, USA
[2] School of Computing, University of Leeds, Leeds, UK

Synonyms

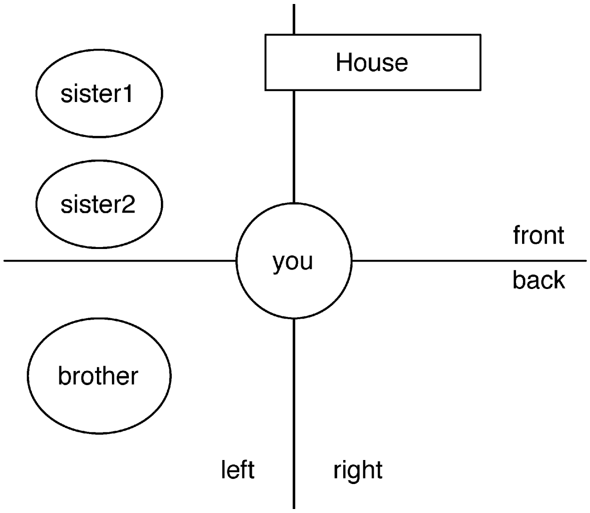
Rough approximation; Rough Set Theory

Definition

Approximations are representations that describe entities in terms of relations to cells in a partition which serves as a frame of reference. Approximations give raise to an indiscernibility relation: In the 'approximation space' two entities are indiscernible if and only if they have identical approximations. Approximations are used as tools for the representation of objects with indeterminate boundaries and multi-resolution spatial, temporal, and attribute data.

Example

At every moment in time, your body-axes create a partition of space consisting of the cells *front-left* (fl), *back-left* (bl), *front-right* (fr), and *back-right* (br) as depicted in Fig. 1. Every object, including *your-brother* (yb), *your-sister1* (ys1), *your-sister2* (ys1), and *your-house* (yh), can be characterized in terms of their relations to the cells of the partition. For example, part-of(ys1,fl), disjoint(ys1,fr), disjoint(ys1,br), disjoint(ys1,bl), partly-overlaps(yh,fl), partly-overlaps(yh,fr), and so on. Two objects are indiscernible with respect to the underlying partition if they have the same mereological relations to all cells of the partition. For example, your two sisters, ys1 and ys2, are *indiscernible* when described in terms of the relations to the cells of the partition P1 = {fl,fr,br,bl} since they have the same relations to the members of P1. Notice, that in a coarser frame of reference more objects become indiscernible. For example, with respect to the partition P2 = {left,right}, all your siblings become indiscernible



Approximation, Figure 1 Approximation in a frame of reference created by your major body axes

(all three are part of *left* and disjoint from *right*). Notice also, that from the facts that partition cells are disjoint and part-of(ys1,fl) and part-of(yb,fb) hold, one can derive that ys1 and yb are disjoint, i. e., the structure of the partition can be taken into account in reasoning processes.

Historical Background

Rough set theory, the formal basis of the theory of approximations as reviewed in this entry, was introduced by Pawlak [11,12] as a formal tool for data analysis. The main areas of application are still data mining and data analysis [7,10,17], however there are successful applications in GIScience [3,21,22] and in other areas. Ongoing research in rough set theory includes research on rough mereology [14] and its application to spatial reasoning [15]. Rough mereology is a generalization of rough set theory and of the research presented here.

Scientific Fundamentals

Rough Set Theory

Rough set theory [10,11,12] provides a formalism for approximating subsets of a set when the set is equipped with an equivalence relation. An equivalence relation is a binary relation which is reflexive, symmetric, and transitive. Given a set $\mathcal{X}$, an equivalence relation on $\mathcal{X}$ creates a partition $\mathcal{I}$ of $\mathcal{X}$ into a set of jointly exhaustive and pairwise disjoint subsets. Let $[x]$ be the set of all members of $\mathcal{X}$ that are equivalent to x with respect to, i. e., $[x] = \{y \in \mathcal{X} \mid x \sim y\}$. Then, $\mathcal{I} = \{[x] \mid x \in \mathcal{X}\}$ is a partition of $\mathcal{X}$: the union of all members of $\mathcal{I}$ is $\mathcal{X}$ and no distinct members of $\mathcal{I}$ overlap.

An arbitrary subset $b \subseteq \mathcal{X}$ can be approximated by a function $\varphi_b : \mathcal{I} \to$ **fo**, **po**, **no**. The value of $\phi_b\,[x]$ is defined to be **fo** if $[x] \subseteq b$, it is **no** if $[x] \cap b = \varnothing$, and otherwise the value is **po**. The three values **fo**, **po**, and **no** stand respectively for 'full overlap', 'partial overlap' and 'no overlap'; they measure the extent to which b overlaps the members of the partition $\mathcal{I}$ of $\mathcal{X}$.

Regional Approximations

In spatial representation and reasoning it is often not necessary to approximate subsets of an arbitrary set, but subsets of a set with *topological* or *geometric* structure. Thus, rather than considering arbitrary sets and subsets thereof, regular closed subsets of the plane are considered. The cells (elements) of the partitions are regular closed sets which may overlap on their boundaries, but not their interiors.

Consider Fig. 2. Let $\mathcal{X} = \{(x, y) \mid 0 < x < 7 \,\&\, 0 < y < 7\}$ be a regular closed subset of the plane and $c_{(0,0)} = \{(x, y) \mid 0 < x < 1 \,\&\, 0 < y < 1\}$, $c_{(0,1)} = \{(x, y) \mid 0 < x < 1 \,\&\, 1 < y < 2\}, \ldots\, c_{(7,7)} = \{(x, y) \mid 6 < x < 7 \,\&\, 6 < y < t\}$, a partition of $\mathcal{X}$ formed by the regular closed sets $c_{(0,0)}, \ldots, c_{(6,6)}$ (cells), i. e., $\mathcal{I} = \{c_{(0,0)}, \ldots, c_{(6,6)}\}$. Two members of $\mathcal{X}$ are equivalent if and only if they are part of the interior of the same cell $c_{(i,j)}$.

The subsets x, y, z, and u now can be approximated in terms of their relations to the cells $c_{(i,j)}$ of $\mathcal{I}$ which is represented by the mappings $\phi_x, \phi_y, \phi_z, \phi_u$ of signature $\mathcal{I} \to \Omega$ with $\Omega =$ **fo**, **po**, **no**):

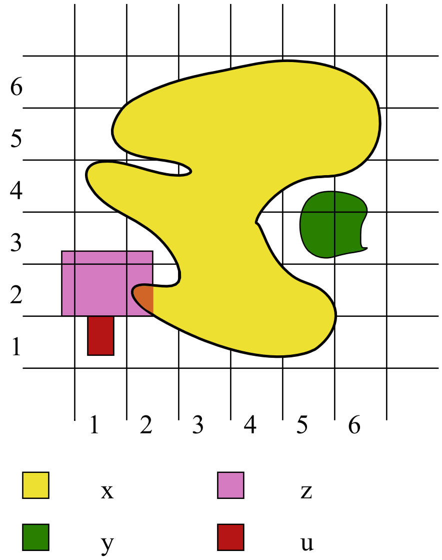
$$X = \varphi_x = \frac{\mathcal{I}}{\Omega}\left\|\begin{array}{c|c|c|c|c|c} \ldots & c_{(2,6)} & c_{(3,6)} & \ldots & c_{(3,5)} & \ldots \\ \hline \ldots & \mathsf{po} & \mathsf{po} & \ldots & \mathsf{fo} & \ldots \end{array}\right.$$

$$Y = \varphi_y = \frac{\mathcal{I}}{\Omega}\left\|\begin{array}{c|c|c|c|c|c} \ldots & c_{(5,4)} & c_{(6,4)} & c_{(5,3)} & c_{(6,3)} & \ldots \\ \hline \ldots & \mathsf{po} & \mathsf{po} & \mathsf{po} & \mathsf{po} & \ldots \end{array}\right.$$

$$Z = \varphi_z = \frac{\mathcal{I}}{\Omega}\left\|\begin{array}{c|c|c|c|c|c} \ldots & c_{(0,1)} & c_{(1,2)} & c_{(2,2)} & c_{(3,2)} & \ldots \\ \hline \ldots & \mathsf{po} & \mathsf{fo} & \mathsf{po} & \mathsf{no} & \ldots \end{array}\right.$$

$$U = \varphi_u = \frac{\mathcal{I}}{\Omega}\left\|\begin{array}{c|c|c|c|c|c} \ldots & c_{(0,1)} & c_{(1,1)} & c_{(1,2)} & c_{(1,3)} & \ldots \\ \hline \ldots & \mathsf{po} & \mathsf{no} & \mathsf{no} & \mathsf{no} & \ldots \end{array}\right.$$

In GIScience, regular closed sets like $\mathcal{X}$, the members of $\mathcal{I}$, as well as x, y, z, u are usually considered to model *crisp regions*, i. e., regions with crisp and well defined boundaries. The mappings $\phi_x, \phi_y, \phi_z, \phi_u$ are called *rough approximations* of the (crisp) regions x, y, z, u with respect to the partition $\mathcal{I}$. In the reminder, the notions 'regular closed set', 'crisp region' and 'region' are used synonymously. Non-capitalized letters are used as variables for regions and capitalized letters are used as variables for approximation mappings, i. e., X is used instead of ϕ_x to refer to the rough approximation of x.



Approximation, Figure 2 Rough approximations of spatial regions [3]

Indiscernibility

Given a partition $\mathcal{I}$ of a regular set $\mathcal{X}$, each of the approximation functions X, Y, Z, U stands for a whole set of regular subsets $\mathcal{R}(\mathcal{X})$ of $\mathcal{X}$. For example, X stands for all sets having the approximation X. This set (of regular sets) will be denoted $[\![x]\!] = \{y \in \mathcal{R}(\mathcal{X}) \mid X = Y\}$. Correspondingly, one can introduce an *indiscernibility* relation $\approx$ between regular subsets of $\mathcal{X}$: x and y are indiscernible with respect to the partition $\mathcal{I}$, $x \approx_{\mathcal{I}} y$, if and only if x and y have the same approximation with respect to $\mathcal{I}$, i. e., $X = Y$. Through this indiscernibility relation, the notion of approximation is closely related to the notion of *granularity* in the sense of [9].

Operations on Approximations

The domain of regions is equipped with a meet operation interpreted as the intersection of regions. In the domain of approximation functions the meet operation between regions is approximated by pairs of greatest minimal, $\wedge_{\min}$, and least maximal, $\wedge_{\max}$, meet operations on approximation mappings [2].

Consider the operations $\underline{\triangle}_{\min}$ and $\underline{\triangle}_{\max}$ on the set $\Omega = \{$**fo**, **po**, **no**$\}$ that are defined as follows.

$\underline{\triangle}_{\min}$	no	po	fo
no	no	no	no
po	no	no	po
fo	no	po	fo

$\underline{\triangle}_{\max}$	no	po	fo
no	no	no	no
po	no	po	po
fo	no	po	fo

These operations extend to elements of $\Omega^{\mathcal{I}}$ (i. e., the set of functions from $\mathcal{I}$ to Ω) by:

$$(X \wedge_{\min} Y)c =_{\text{def}} (Xc) \triangle_{\min} (Yc)$$
$$(X \wedge_{\max} Y)c =_{\text{def}} (Xc) \triangle_{\max} (Yc) .$$

In the example it holds that $(Z \wedge_{\min} X)c_{(2,2)} = \textsf{no}$ and $(Z \wedge_{\max} X)c_{(2,2)} = \textsf{po}$ since $(Zc_{(2,2)}) = \textsf{po}$, $(Zc_{(2,2)}) = \textsf{po}$, $\textsf{po} \triangle_{\min} \textsf{po} = \textsf{no}$ and $\textsf{po} \triangle_{\max} \textsf{po} = \textsf{po}$.

Reasoning About Approximations

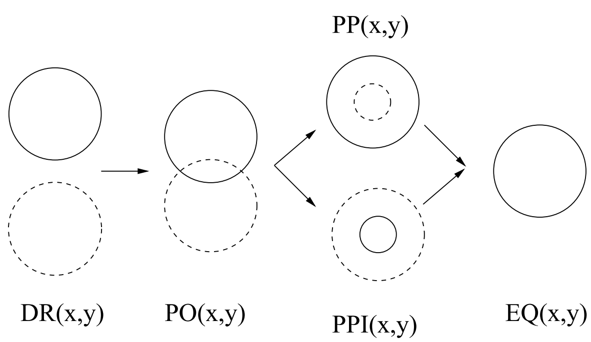
Consider the RCC5 relations [16] disjoint (**DR**), partial overlap (**PO**), proper part (**PP**), has proper part (**PPi**) and equal (**EQ**) between two regions as depicted in Fig. 3. Given two regions x and y these relation between them can be determined by considering the triple of boolean values:

$$(x \wedge y \neq \perp,\ x \wedge y = x,\ x \wedge y = y) .$$

The correspondence between such triples and the relations **DR**, **PO**, **PP**, **PPi**, and **EQ** are given in Table 1. Notice that these definitions of RCC5 relations are exclusively formulated in terms of statements about meet operations (intersections).
The set of triples is partially ordered by setting

$$(a_1, a_2, a_3) < q(b_1, b_2, b_3) \text{ iff } a_i < qb_i \quad \text{for } i = 1, 2, 3$$

where the Boolean values are ordered by **F** < **T**. The resulting ordering (which is similar to the conceptual neighborhood graph [8]) is indicated by the arrows in Fig. 3.



Approximation, Figure 3 RCC5 relations with ordering relations

Approximation, Table 1 Definition of RCC5 relations exclusively using the meet operator [2]

$x \wedge y \neq \perp$	$x \wedge y = x$	$x \wedge y = y$	RCC5
F	F	F	DR
T	F	F	PO
T	T	F	PP
T	F	T	PPi
T	T	T	EQ

There are two approaches one can take to generalize the RCC5 classification from precise regions to approximations of regions. These two may be called the semantic and the syntactic.
Semantic generalization. One can define the RCC5 relationship between approximations X and Y to be the set of relationships which occur between any pair of precise regions having the approximations X and Y. That is, one can define

$$SEM(X, Y) = \{RCC5(x, y) \mid x \in [\![X]\!] \text{ and } y \in [\![Y]\!]\} .$$

Syntactic generalization. One can take a formal definition of RCC5 in the precise case which uses meet operations between regions and generalize this to work with approximations of regions by replacing the meet operations on regions by analogous ones for approximations.
If X and Y are approximations of regions (i. e., functions from $\mathcal{I}$ to Ω), one can consider the two triples of Boolean values:

$$(X \wedge_{\min} Y \neq \perp,\ X \wedge_{\min} Y = X,\ X \wedge_{\min} Y = Y) ,$$
$$(X \wedge_{\max} Y \neq \perp,\ X \wedge_{\max} Y = X,\ X \wedge_{\max} Y = Y) .$$

In the context of approximations of regions, the bottom element, $\perp$, is the function from $\mathcal{I}$ to Ω which takes the value **no** for every element of $\mathcal{I}$. Each of the above triples provides an RCC5 relation, thus the relation between X and Y can be measured by a pair of RCC5 relations. These relations will be denoted by $R_{\min}(X,Y)$ and $R_{\max}(X,Y)$. One then can prove that the pairs $(R_{\min}(X,Y), R_{\max}(X,Y))$, which can occur, are all pairs (a,b) where $a \leq b$ with the exception of (**PP**,**EQ**) and (**PPi**, **EQ**).
Let the syntactic generalization of RCC5 defined by

$$SYN(X, Y) = (R_{\min}(X, Y), R_{\max}(X, Y)) ,$$

where $R_{\min}$ and $R_{\max}$ are defined as described in the previous paragraph. It then follows that for any approximations X and Y, the two ways of measuring the relationship of X to Y are equivalent in the sense that

$$SEM(X, Y) = \{\rho \in \text{RCC5} \mid R_{\min}(X, Y) < q\rho < qR_{\max}(X, Y)\} ,$$

where RCC5 is the set {**EQ**, **PP**, **PPi**, **PO**, **DR**}, and $\leq$ is the ordering as indicated by the arrows in Fig. 3.

Key Applications

The theoretical framework of rough approximations presented above has been applied to geographic informations in various ways, including spatial representation of objects with indeterminate boundaries, representation of spatial data at multiple levels of resolution, representation of attribute data at multiple levels of resolution, and the representation of temporal data.

Objects with Indeterminate Boundaries

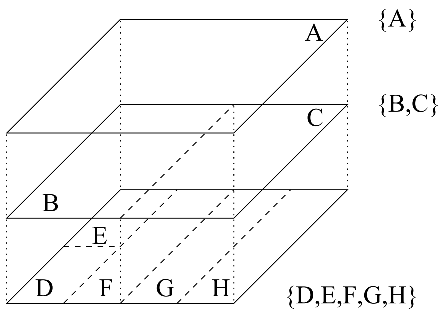
Geographic information is often concerned with natural phenomena, cultural, and human resources. These domains are often formed by objects with indeterminate boundaries [5] such as 'The Ruhr', 'The Alps', etc. Natural phenomena, cultural, and human resources are not studied in isolation. They are studied in certain contexts. In the spatial domain, context is often provided by regional partitions forming frames of reference. Consider, for example, the location of the spatial object 'The Alps'. It is impossible to draw exact boundaries for this object. However, in order to specify its location, it is often sufficient to say that parts of 'The Alps' are located in South Eastern France, Northern Italy, Southern Germany, and so on. This means that one can specify the rough approximation of 'The Alps' with respect to the regional partition created by the regions of the European states. This regional partition can be refined by distinguishing northern, southern, eastern, and western parts of countries. It provides a frame of reference and an ordering structure which is used to specify the location of 'The Alps', and which can be exploited in the representation and reasoning process as demonstrated above.

The utilization of rough approximations in the above context allows one to separate two aspects: (a) The *exact* representation of the location of well defined objects using crisp regions, and (b) the finite *approximation* of the location of objects with indeterminate boundaries in terms of their relations to the regions of the well defined ones. The approximation *absorbs* the indeterminacy [3] and allows for determinate representation and reasoning techniques as demonstrated above.

An important special case is the approximation of objects with indeterminate boundaries with respect to so-called *egg-yolk* partitions [6]. Here the partition consists of three concentric discs, called the central core, the broad boundary, and the exterior. An egg-yolk partition is chosen such that an object with indeterminate boundaries has the relation **fo** to the central core, the relation **po** to the broad boundary, and the relation **no** to the exterior cell of the partition.

Processing Approximate Geographic Information at Multiple Levels of Detail

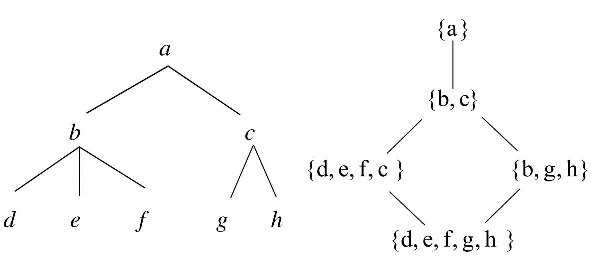
Partitions that form frames of references for rough approximations can be organized hierarchically. In Fig. 4, three partitions which partition the region A at different levels of resolution are depicted: $\{A\}$, $\{B,C\}$, $\{D,E,F,G,H\}$. Obviously, parts/subsets of A can be approximated at different levels of granularity with respect to $\{A\}$, $\{B,C\}$, or $\{D,E,F,G,H\}$. Various approaches of processing approximations at and across different levels of granularity in such hierarchical subdivisions have been proposed including [4,20,21,22].



Approximation, Figure 4 Partitions at multiple levels of resolution [4]

Processing Attribute Data

From the formal development of rough approximations, it should be clear that its application is not limited to the approximation of spatial location, but can be applied in the same way to attribute data. For example, at a coarse level of representing attribute data one might ignore the distinction between different kinds of roads (motorways, major roads, minor roads, etc.) and represent only a single class 'road'. Consider the classification tree in the left part of Fig. 5. One can create 'partitions' of the class a (sets of jointly exhaustive and pairwise disjoint subclasses of a) at different levels of resolutions as indicated in the lattice in the right part of the figure. Let a–h be subsets of a, then other subsets of a can be approximated with respect to the various partitions in the ways described above. Again, see also [4,19,20,21,22].



Approximation, Figure 5 A classification tree (*left*), the corresponding lattice of possible partitions of the root class [4]

Temporal Data

Humans have sophisticated calendars that hierarchically partition the time-line in different ways, for example, into minutes, hours, days, weeks, months, etc. Representations and reasoning about the temporal location of events and processes need to take into account that events and processes often lie skew to the cells of calendar partitions (i. e., 'x happened yesterday' does not mean that x started at 12 a.m. and ended at 0 p.m.) Thus, descriptions of the temporal location of events and processes are often approximate and rough in nature rather than exact and crisp. As demonstrated in [1] and [18], rough approximation and reasoning methods of the sort introduced above can be used to represent and to reason about approximate temporal location.

Cross References

► Representing Regions with Indeterminate Boundaries
► Uncertainty, Semantic

Recommended Reading

1. Bittner, T.: Approximate qualitative temporal reasoning. Annals of Mathematics and Artificial Intelligence **35**(1–2), 39–80 (2002)
2. Bittner, T., Stell, J.G.: Approximate qualitative spatial reasoning. Spatial Cognition and Computation **2**(4),435–466 (2002)
3. Bittner, T., Stell, J.G.: Vagueness and rough location. GeoInformatica **6**, 99–121 (2002)
4. Bittner, T., Stell, J.G.: Stratified rough sets and vagueness. In: Kuhn, W., Worboys, M., Impf, S. (eds.) Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science. International Conference COSIT'03, pp. 286–303. Springer, Berlin (2003)
5. Burrough, P., Frank, A.U. (eds.): Geographic Objects with Indeterminate Boundaries, GISDATA Series II. Taylor and Francis, London (1995)
6. Cohn, A.G., Gotts, N.M.: The 'egg-yolk' representation of regions with indeterminate boundaries. In: Burrough, P.A., Frank, A.U. (eds.) Geographic Objects with Indeterminate Boundaries, GISDATA Series II. pp. 171–187. Taylor and Francis, London (1996)
7. Duentsch, I., Gediga, G.: Rough set data analysis: A road to non-invasive knowledge discovery. Methodos Publishers, Bangor (2000)
8. Goodday, J.M., Cohn,A.G.: Conceptual neighborhoods in temporal and spatial reasoning. In: ECAI-94 Spatial and Temporal Reasoning Workshop (1994)
9. Hobbs, J.: Granularity. In: Proceedings of the IJCAI 85 (1985)
10. Orłowska, E.(ed.): Incomplete Information – Rough Set Analysis. Studies in Fuzziness and Soft Computing, vol.13 Physica-Verlag, Heidelberg (1998)
11. Pawlak, Z.: Rough sets. Internat. J. Comput. Inform. **11**, 341–356 (1982)
12. Pawlak, Z.: Rough sets: theoretical aspects of reasoning about data. Theory and decision library. Series D, System theory, knowledge engineering, and problem solving, vol. 9. Kluwer Academic Publishers, Dordrecht, Boston (1991)
13. Pawlak, Z., Grzymala-Busse, J., Slowinski, R., Ziarko, R.A.: Rough sets. Communications of the ACM **38**(11), 89–95 (1995)
14. Polkowski, L., Skowron, A.: Rough mereology: A new paradigm for approximate reasoning. J. Approx. Reason. **15**(4),333–365 (1996)
15. Polkowski, L.: A survey of recent results on spatial reasoning via rough inclusions. In: Bolc, L., Michalewicz, Z., Nishida, T. (eds.) Intelligent Media Technology for Communicative Intelligence, Second International Workshop, IMTCI 2004, Lecture Notes in Computer Science, Springer, Berlin (2004)
16. Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. In: Nebel, B., Rich, C., Swartout, W. (eds.) Principles of Knowledge Representation and Reasoning. Proceedings of the Third International Conference (KR92), pp. 165–176. Morgan Kaufmann (1992)
17. Slezak, D., Wang, G., Szczuka, M.S., Düntsch, I., Yao, Y. (eds.) Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, 10th International Conference RSFDGrC 2005 (Part I), Lecture Notes in Computer Science. vol. 3641 Aug 31–Sept 3, Springer, Regina, Canada (2005)
18. Stell, J.G.: Granularity in change over time. In: Duckham, M., Goodchild, M., Worboys, M. (eds.) Foundations of Geographic Information Science, pp. 95–115. Taylor and Francis (2003)
19. Stell, J.G.: The representation of discrete multi-resolution spatial knowledge. In: Cohn, A.G., Giunchiglia, F., Selman, B. (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference (KR2000), pp. 38–49. Morgan Kaufmann (2000)
20. Stell, J.G., Worboys, M.F.: Stratified map spaces: A formal basis for multi-resolution spatial databases. In: Poiker, T.K., Chrisman, N. (eds.) Proceedings 8th International Symposium on Spatial Data Handling SDH'98, pp. 180–189. International Geographical Union (1998)
21. Worboys, M.F.: Computation with imprecise geospatial data. Computers, Environment and Urban Systems **22**, 85–106 (1998)
22. Worboys, M.F.: Imprecision in finite resolution spatial data. GeoInformatica **2**, 257–279 (1998)

aR-Tree

► Multi-Resolution Aggregate Tree

ArcExplorer

► Web Feature Service (WFS) and Web Map Service (WMS)

ArcGIS: General Purpose GIS Software System

David J. Maguire
ESRI, Redlands, CA, USA

Synonyms

GIS software; ESRI

Definition

ArcGIS is a general purpose GIS software system developed by ESRI. It is an extensive and integrated software platform technology for building operational GIS. ArcGIS comprises four key software parts: a geographic information model for modeling aspects of the real world; components for storing and managing geographic information in files and databases; a set of out-of-the-box applications for creating, editing, manipulating, mapping, analyzing and disseminating geographic information; and a collection of web services that provide content and capabilities (data and functions) to networked software clients. Parts of the ArcGIS software system can be deployed on mobile devices, laptop and desktop computers and servers.

From the end user perspective ArcGIS has very wide ranging functionality packaged up into a generic set of menu-driven GIS applications that implement key geographic workflows. The applications deal with geographic data creation, import and editing, data integration and management, data manipulation and organization, and data analysis, mapping and reporting. Additionally, ArcGIS On-line provides a set of web services that can be access from any web-enabled device, browser or other application.

ArcGIS is also a developer-friendly product. The software is accessible to developers using several programming paradigms including within application scripting (Python, VBScript and JScript), web services end points (SOAP/XML, KML), and as component interfaces (.Net and Java). Developers can personalize and customize the existing software applications, build whole new applications, embed parts of ArcGIS in other software, and interface to other software systems.

Historical Background

ArcGIS is developed by a company called Environmental Systems Research Institute, Inc (ESRI – pronounce each letter, it is not an acronym). Headquartered in Redlands, California and with offices throughout the world, ESRI was founded in 1969 by Jack and Laura Dangermond (who to this day are President and Vice-President) as a privately held consulting firm that specialized in land use analysis projects. The early mission of ESRI focused on the principles of organizing and analyzing geographic information; projects included developing plans for rebuilding the City of Baltimore, Maryland, and assisting Mobil Oil in selecting a site for the new town of Reston, Virginia.

During the 1980s ESRI devoted its resources to developing and applying a core set of application tools that could be applied in a computer environment to create a geographic information system. In 1982 ESRI launched its first commercial GIS software called ARC/INFO. It combined computer display of geographic features, such as points, lines, and polygons (the ARC software), with a database management tool for assigning attributes to these features (the Henco Inc. INFO DBMS). Originally designed to run on minicomputers, ARC/INFO was the first modern GIS software system. As the technology shifted operating system, first to UNIX and later to Windows, ESRI evolved software tools that took advantage of these new platforms. This shift enabled users of ESRI software to apply the principles of distributed processing and data management.

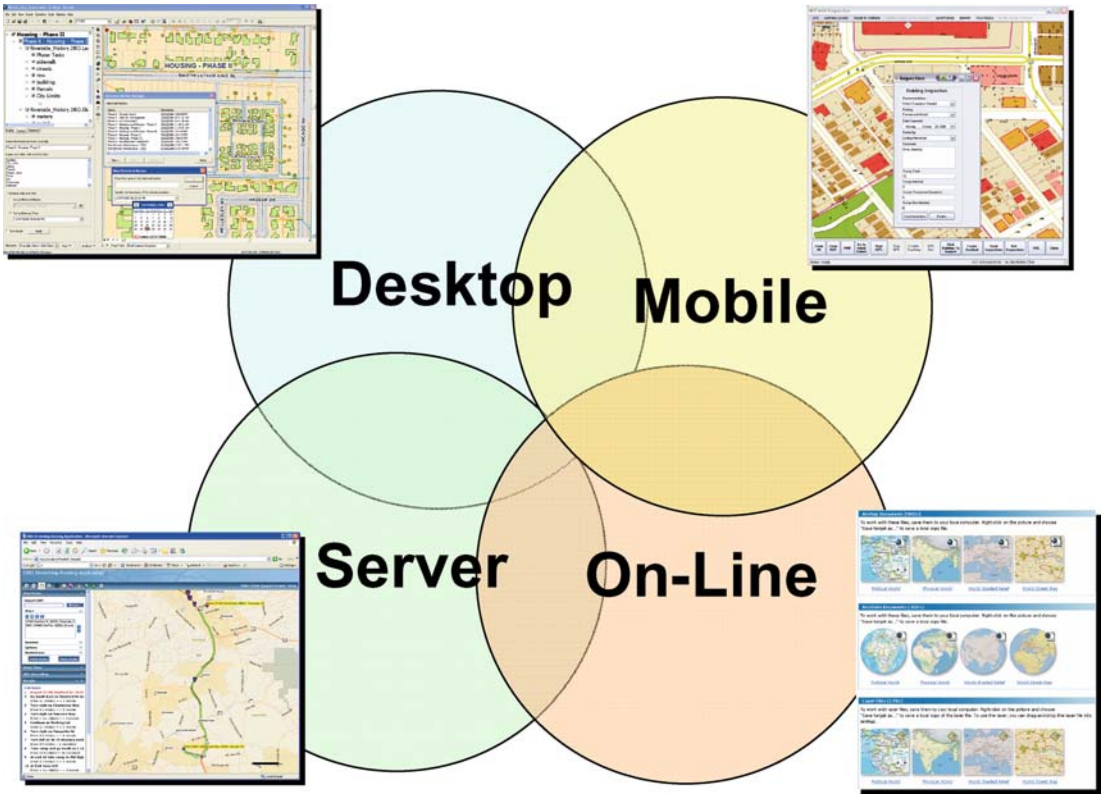
The 1990s brought more change and evolution. The global presence of ESRI grew with the release of ArcView, an affordable, relatively easy-to-learn desktop mapping tool, which shipped 10,000 copies in the first six months of 1992. In the mid-1990s ESRI released the first of a series of Internet based map servers that published maps, data, and metadata on the web. These laid the foundation for today's server-based GIS called ArcGIS Server and a suite of on-line web services called ArcWeb Services.

In 1997 ESRI embarked on an ambitious research project to reengineer all of its GIS software as a series of reusable software objects. Several hundred person-years of development later ArcInfo 8 was released in December 1999. In April 2001, ESRI began shipping ArcGIS 8.1, a family of software products that formed a 'complete' GIS built on industry standards that provides powerful, yet easy-to-use capabilities right out of the box. ArcGIS 9 followed in 2003 and saw the addition of ArcGIS Server and ArcGIS On-line – a part of ArcGIS that ESRI's hosts in its own servers and makes accessible to users over the web.

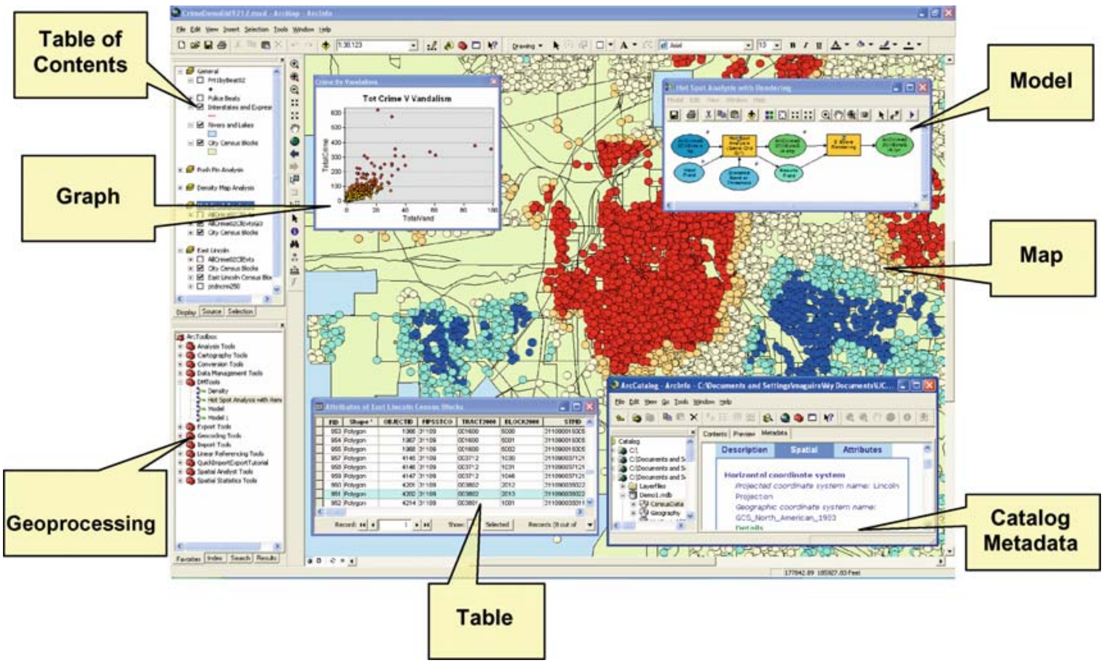
Although developed as a complete system, ArcGIS 9 is a portfolio of products and is available in individual parts. The major product groups are desktop, server, on-line and mobile (Fig. 1). ArcGIS desktop has a scalable set of products, in increasing order of functionality, ArcReader, ArcView, ArcEditor and ArcInfo (Fig. 2). ESRI has built plug-in extensions (3D Analyst, Spatial Analyst, Network Analyst, etc.) which add new functional capabilities to the main desktop products. There is a desktop run-time called ArcGIS Engine which is a set of software components that developers can use to build custom applications and embed GIS functions in other applications. ArcGIS Server is also a scalable set of products, namely, ArcGIS Server Basic, Standard and Advanced (with each available in either workgroup or enterprise editions). The mobile products include ArcPad and ArcGIS Mobile, and to complete the picture there are a suite of ArcGIS On-line web services which provide data and applications to desktop, server and mobile clients.

Today, ESRI employs more than 4,000 staff worldwide, over 1,900 of which are based at the worldwide headquarters in California. With 27 international offices, a network of more than 50 other international distributors, and over 2000 business partners, ESRI is a major force in the GIS industry. ESRI's lead software architect, Scott Morehouse remains the driving force behind ArcGIS development and he works closely with Clint Brown, product development

A



ArcGIS: General Purpose GIS Software System, Figure 1 ArcGIS platform



ArcGIS: General Purpose GIS Software System, Figure 2 ArcGIS desktop user interface director, David Maguire, product director, and, of course, Jack Dangermond, president.

Scientific Fundamentals

Fundamental Functional Capabilities

ArcGIS is a very big software system with literally thousands of functional capabilities and tens of millions of lines of software code. It is impossible, and in any case worthless, to try to describe each piece of functionality here. Instead, the approach will be to present some of the core foundational concepts and capabilities.

The best way to understand ArcGIS is to start with the core information (some people use the term data) model since it is this which defines what aspects of the world can be represented in the software and is the push off point for understanding how things can be manipulated. ArcGIS's core information model is called the geographic database, or geodatabase for short. The geodatabase defines the conceptual and physical model for representing geographic objects and relationships within the system. Geodatabases work with maps, models, globes, data and metadata. Instantiated geodatabases comprise information describing geographic objects and relationships that are stored in files or DBMS. These are bound together at run-time with software component logic that defines and controls the applicable processes. It is this combination of data (form) and software (process) which makes the geodatabase object-oriented and so powerful and useful. For example, a geodatabase can represent a linear network such as an electricity or road network. The data for each link and node in a network is stored as a separate record. Functions (tools or operators), such as tracing and editing that work with networks access all the data together and organize it into a network data structure prior to manipulation. Geodatabases can represent many types of geographic objects and associated rules and relationships including vector features (points, lines, polygons, annotations [map text], and 3D multipatches), rasters, addresses, CAD entities, topologies, terrains, networks, and surveys. In ArcGIS, geographic objects of the same type (primarily the same spatial base – dimensionality, projection, etc.) are conventionally organized into a data structure called a layer. Several layers can be integrated together using functions such as overlay processing, merge and map algebra. Geodatabases can be physically stored both in file system files and DBMS tables (e. g. in DB2, Oracle and SQL Server).

It is convenient to discuss the functional capabilities of ArcGIS in three main categories: geovisualization, geoprocessing and geodata management.

Geovisualization, as the name suggests, is concerned with the visual portrayal of geographic information. It should come as no surprise that many people frequently want to visualize geographic information in map or chart form. Indeed many people's primary use for a GIS is to create digital and/or paper maps. ArcGIS has literally hundreds of functions for controlling the cartographic appearance of maps. These include specifying the layout of grids, graticules, legends, scale bars, north arrows, titles, etc., the type of symbolization (classification, color, style, etc.) to be used, and also the data content that will appear on the final map. Once authored, maps can printed or published in softcopy formats such as PDF, or served up over the web as live map services. Additionally, many geographic workflows are best carried out using a map-centric interface. For example, editing object geometries, examined the results of spatial queries, and verifying the results of many spatial analysis operations can only really be performed satisfactorily using a map-based interface. ArcGIS supports multiple dynamic geovisualization display options such as 2D geographic (a continuous view of many geodatabase layers), 2D layout (geodatabase layers presented in 'paper' space), 3D local scenes (strictly a 2.5D scene graph view of local and regional data) and 3D global (whole Earth view with continuous scaling of data).

The term 'geoprocessing' is used to describe the spatial analysis and modeling capabilities of ArcGIS. ArcGIS adopts a data transformation framework approach to analysis and modeling: data + operator = data. For example, *streets* data + *buffer* operator = *streets_with_buffers* data. ArcGIS has both a framework for organizing geoprocessing and an extensive set of hundreds of operators that can be used to transform data. The framework is used to organize operators (also called functions or tools), compile and execute geoprocessing tasks or models (collections of tools and data organized as a work flow), and interfaces to the other parts of ArcGIS that deal with geodata management and geovisualization. The set of operators includes tools for 'classic' GIS analysis (overlay, proximity, etc.), projection/coordinate transformation, data management and conversion, domain specific analysis – 3D, surfaces, network, raster, geostatistics, linear referencing, cartography, etc, and simulation modeling. Geoprocessing is widely used to automate repetitive tasks (e. g. load 50 CAD files into a geodatabase), integrate data (e. g. join *major_streets* and *minor_streets* data layers to create a single *complete_streets* layer), as part of quality assurance work flows (e. g. find all buildings that overlap), and to create process models (e. g. simulate the spread of fire through a forested landscape).

Geodata management is a very important part of GIS not least because geodata is a very valuable and critical component of most well-established operational GIS. It is especially important in large enterprise GIS implementations because the data volumes tend to be enormous, and multiple users often want to share access. ArcGIS has responded to these challenges by developing advanced technology to store and manage geodata in databases and files. An efficient storage schema, and well-tuned spatial and attribute indexing mechanisms, support rapid retrieval of data record sets. Coordinating multi-user updates to continuous geographic databases has been a thorny problem for GIS developers for many years. ArcGIS addresses this using an optimistic concurrency strategy based on versioning. The versioning data management software, data schema and application business logic are a core part of ArcGIS. The data in ArcGIS can be imported/exported in many standard formats (e. g. dxf and mif), is accessible via standards-based interfaces (e. g. OGC WMS and WFS) and open APIs (application programming interfaces; e. g. SQL, .Net and Java), and the key data structure formats are openly published (e. g. shapefile and geodatabase).

Fundamental Design Philosophy

The ArcGIS software has evolved considerably over the two and a half decades of its existence as the underlying computer technologies, and concepts and methods of GIS have advanced. Nevertheless, many of original design philosophies are still cornerstones of each new release. Not surprisingly, the original design goals have been supplemented by more recent additions which today drive the software development process. This section discusses the fundamental design philosophies of ArcGIS in no particular order of significance.

Commercial off-the-shelf (COTS) hardware. ArcGIS has always run on industry standard COTS hardware platforms (including computers and associated peripherals, such as digitizers, scanners and printers). Today, hardware is insulated by a layer of operating system software (Windows, Linux, Solaris, etc.) and this constitutes much of the underlying ‘computing platform’ on which the GIS software runs. The operating system affords a degree of hardware neutrality. ArcGIS runs on well established mainstream operating systems and hardware platforms.

Multiple computer architectures. Parts of the ArcGIS software system can run on desktop, server, and mobile hardware. There is also a portion of ArcGIS that is available on-line for use over the web. The software can be configured to run stand alone on desktop and mobile machines. It can also be configured for workgroup and enterprise use so that it runs as a client-server, and/or distributed server-based implementation. This offers considerable flexibility for end use deployment. The newest release of the software is adept at exploiting the web as a platform for distributed solutions.

GIS professionals. The target user for the core of ArcGIS is the GIS professional (loosely defined as a career GIS staff person). GIS professionals often build and deploy professional GIS applications for end users (for example, planners, utility engineers, military intelligence analysts, and marketing staff). The software is also frequently incorporated in enterprise IT systems by IT professionals, and is increasingly being used by consumers (members of the general public with very limited GIS skills).

Generic tool box with customization. From the outset ArcGIS was designed as a tool box of generic GIS tools. This means that functional GIS capabilities are engineered as self-contained software components or tools that can be applied to many different data sets and application work flows. This makes the software very flexible and easily adaptable to many problem domains. The downside to this is that the tools need to be combined into application solutions that solve problems, and this adds a degree of complexity. In recent releases of the software this issue has been ameliorated by the development of menu-driven applications for key geographic workflows (editing, map production, 3D visualization, business analysis, utility asset management and design, etc.).

Strong release control. ArcGIS is a software **product** which means that it has well-defined capabilities, extensive on-line help and printed documentation, and add-on materials (third party scripts, application plug-ins, etc.), a license agreement that controls usage, and that it is released under carefully managed version control. This means that additions and updates to the product are added only at a new release (about two to three times a year).

Internationalized and localized. The core software is developed in English and is internationalized so that it can be localized into multiple locales (local languages, data types, documentation, data, etc.). The latest release of ArcGIS has been localized into more that 25 locale languages including Farsi, French, German, Hebrew, Japanese, Italian, Mandarin, Spanish, and Thai.

Key Applications

ArcGIS has been applied to thousands of different application arenas over the years. It is testament to the software’s flexibility and adaptability that it has been employed in so many different application areas. By way of illustration this section describes some example application areas in which ArcGIS has been widely adopted.

Business

Businesses use many types of information – geographic locations, addresses, service boundaries, sales territories, delivery routes and more that can be viewed and analyzed in map form. ArcGIS software integrated with business, demographic, geographic, and customer data produces applications that can be shared across an entire organization. Typical applications include selecting the best sites, profiling customers, analyzing market areas, updating and managing assets in real time, and providing location-based services (LBS) to users. These applications are used extensively in banking and financial services, retailing, insurance, media and press and real estate sectors.

Education

In the education sector ArcGIS is applied daily in administration, research and teaching at the primary, secondary and tertiary levels. In recent years, ArcGIS use has grown tremendously, becoming one of the hottest new research and education tools. At the primary and secondary level GIS provides a set life skills and a stimulating learning environment. More than 100 higher education academic disciplines have discovered the power of spatial analysis with GIS. Researchers are using GIS to find patterns in drug arrests, study forest rehabilitation, improve crop production, define urban empowerment zones, facilitate historic preservation, develop plans to control toxic waste spills, and much more. GIS is also a useful tool for the business of education. It is used to manage large campuses, plan campus expansion, and provide emergency campus response plans. It is also used by administrators to track graduates and alumni or identify from where potential new students may be recruited.

Government

Government organizations throughout the world are under increasing pressure to improve services and streamline business practices while adhering to complex political or regulatory requirements. To do so, they must digest huge amounts of information, most of which is tied to a very specific geographic location – a street, an address, a park, a piece of land. As a result, ArcGIS has become indispensable for most large and many small governments. The applications of ArcGIS are very diverse and their implementation extremely extensive. Very many major government organizations at the national, state, regional and local levels use ArcGIS. Some of the main application areas include: economic development, elections, national policy formulation, homeland security, land records and cadastral solutions, law enforcement, public safety, public works, state and local, sustainable development, and urban and regional planning.

Military

Although ArcGIS is deployed as a niche tool in some application domains, there is increasing realization that enterprise ArcGIS implementations are providing defense-wide infrastructures, capable of supporting fighting missions, command and control, installation management, and strategic intelligence. GIS plays a critical role within the defense community in the application areas of: command and control (C2), defense mapping organizations, base operations and facility management, force protection and security, environmental security and resource management, health and hygiene intelligence, surveillance and reconnaissance systems, logistics, military engineering, mine clearance and mapping, mission planning, peacekeeping operations, modeling and simulation, training, terrain analysis, visualization, and chemical, biological, radiological, nuclear, and high explosive (CBRNE) incident planning and response.

Natural Resources

Just as ArcGIS is routinely used in managing the built environment, it is also very popular in measuring, mapping, monitoring and managing the natural environment. Again the application areas are very wide ranging extending from agriculture, to archaeology, environmental management, forestry, marine and coast, mining and earth science, petroleum and water resources. ArcGIS provides a strong set of tools for describing, analyzing, and modeling natural system processes and functions. Interactions and relationships among diverse system components can be explored and visualized using the powerful analytical and visualization tools that GIS software provides.

Utilities

Utilities (electric, gas, pipeline, telco and water/wastewater) were among the first users of GIS. Today ArcGIS is involved in many of the core activities of utilities including: asset information, business for utilities, network design, emergency management, electricity generation and transmission, land management, outage management, pipeline management, and work force productivity. ArcGIS is used to manage the flow of water and wastewater to service homes and businesses, to track the location and condition of water mains, valves, hydrants, meters, storage facilities, sewer mains, and manholes. The same systems make keeping up with regulatory compliance, TV inspection data, and condition ratings easier. Competitive pressure and regulatory constraints are placing increasing demands on pipeline operators to operate in an efficient and responsible manner. Responding to these demands requires accessibility to information regarding geographically distributed assets and operations. ArcGIS is enabling telecommunication professionals to integrate location-based data into analysis and management processes in network planning and operations, marketing and sales, customer care, data management, and many other planning and problem-solving tasks.

Future Directions

ArcGIS is in a constant state of evolution and even though the core values and capabilities are well established, there is always a need for improvement and expansion into new application areas. While there is new development in all areas, the research agenda currently is centered on the following key topics:

ArcGIS On-line. ArcGIS On-line is a suite of web-based applications that combine data and functionality in a way that supplements the desktop, server and mobile software which is installed on computers in user organizations. The web services include framework and coverage data typically at global and regional scales (e. g., global imagery and street centerline files), and several functional services (e. g. geocoding and routing). Initially released with ArcGIS 9.2, the on-line services are undergoing considerable enhancement in both the 2D and 3D domains.

ArcGIS Mobile. ArcGIS has included mobile capabilities for several releases. The current development focus is on enhancing the mobile capabilities to support the deployment of professional mobile applications by end users. A central piece of this effort is the development of a GIS server that is responsible for data management and running central applications (e. g. mapping, geocoding and routing). There are also clients for several hardware devices including smartphones, and web browsers.

Distributed GIS. In keeping with the general progress of building ArcGIS using industrial strength IT standard technologies, much is being done to make it possible to integrate the GIS software into enterprise information systems and thus distribute GIS throughout an organization. This includes additional work on standards (both GIS domain specific and general IT), web services (especially XML), security (for single sign on authentication), and integration APIs (such as SQL, .Net and Java).

Ease of Use. A key goal of future work is the continued improvement in ease of use. ArcGIS has been feature rich for many releases, but a little daunting for new users. A new desktop user interface design and careful attention to user workflows, combined with improvements in interactive performance should go someway to satisfying the requirements of usability.

Cross References

► Distributed Geospatial Computing (DGC)
► Information Services, Geography
► Internet GIS
► Web Services, Geospatial

Recommended Reading

ESRI: What is ArcGIS 9.2? Redlands, California (2006)

Hoel, E., Menon, S., Morehouse, S.: Building a Robust Relational Implementation of Topology. In: Proceedings of 8th International Symposium on Spatial and Temporal Databases – SSTD'03 (2003)

Morehouse, S.: ARC/INFO: a georelational model for spatial information. In: Proceedings of AUTOCARTO 8, pp. 388–97. ASPRS, Falls Church, Virginia (1985)

Morehouse, S.: The ARC/INFO Geographic Information System. Computers Geosci. **18**(4), 435–443 (1992)

Ormsby, T., Napoleon, E., Burke, R., Feaster, L., Groessl, C.: Getting to Know ArcGIS Desktop Second Edition: Basics of ArcView, ArcEditor, and ArcInfo. ESRI Press, Redlands (2004)

Zeiler, M.: Modeling our world: the ESRI guide to database design. ESRI, Redlands, California (1999)

ESRI web site. http://www.esri.com

ArcIMS

► Web Feature Service (WFS) and Web Map Service (WMS)

Arrival, Angle Of

► Indoor Localization

Arrival, Time Of

► Indoor Localization

Artificial Neural Network

► Self Organizing Map (SOM) Usage in LULC Classification

Association

► Co-location Patterns, Algorithms

Association Measures

► Co-location Patterns, Interestingness Measures

Association Rules: Image Indexing and Retrieval

► Image Mining, Spatial

Association Rules, Spatio-temporal

► Movement Patterns in Spatio-temporal Data

Atlas, Electronic

► Multimedia Atlas Information Systems

Atlas Information System

► Multimedia Atlas Information Systems

Atlas, Interactive

► Multimedia Atlas Information Systems

Atlas, Multimedia

► Multimedia Atlas Information Systems

Atlas, Virtual

► Multimedia Atlas Information Systems

Attribute and Positional Error in GIS

► Uncertain Environmental Variables in GIS

Autocorrelation, Spatial

CHANDANA GANGODAGAMAGE[1],
XIAOBO ZHOU[2], HENRY LIN[2]
[1] Saint Anthony Falls Laboratory,
Department of Civil Engineering,
University of Minnesota, Minneapolis, MN, USA
[2] Department of Crop and Soil Sciences,
The Pennsylvania State University,
University Park, PA, USA

Synonyms

Spatial correlation; Spatial dependence; Spatial inter-dependence

Definition

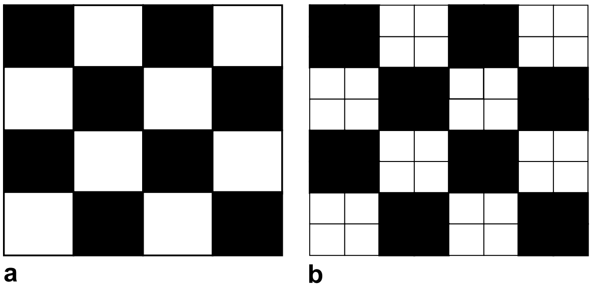
In many spatial data applications, the events at a location are highly influenced by the events at neighboring locations. In fact, this natural inclination of a variable to exhibit similar values as a function of distance between the spatial locations at which it is being measured is known as spatial dependence. Spatial autocorrelation is used to measure this spatial dependence. If the variable exhibits a systematic pattern in its spatial distribution, it is said to be spatially autocorrelated. The existence and strength of such interdependence among values of a specific variable with reference to a spatial location can be quantified as a positive, zero, or negative spatial autocorrelation. Positive spatial autocorrelation indicates that similar values or properties tend to be collocated, while negative spatial autocorrelation indicates that dissimilar values or properties tend to be near each other. Random patterns indicate zero spatial autocorrelation since independent, identically distributed random data are invariant with regard to their spatial location.

Historical Background

The idea of spatial autocorrelation is not new in the literature and was conceptualized as early as 1854, when nebula-like spatial clusters with distance-decay effects were readily apparent in mapped cholera cases in the city of London [1]. This led to the hypothesis that the systematic spatial pattern of Cholera outbreak decayed smoothly with distance from a particular water supply which acted as the source for the disease. This concept of spatial autocorrelation was also documented in the first law of geography in 1970 which states: "Everything is related to everything else, but near things are more related than distant things" [11].

Scientific Fundamentals

Spatial autocorrelation is a property of a variable that is often distributed over space [9]. For example, land surface elevation values of adjacent locations are generally quite similar. Similarly, temperature, pressure, slopes, and rainfall vary gradually over space, thus forming a smooth gradient of a variable between two locations in space. The propensity of a variable to show a smooth gradient across space aggregates similar values or properties adjacent to each other.
In classical statistics, the observed samples are assumed to be independent and identically distributed (iid). This assumption is no longer valid for inherently spatially autocorrelated data. This fact suggests that classical statistical tools like linear regression are inappropriate for spatial



Autocorrelation, Spatial, Figure 1 The strength of spatial autocorrelation as a function of scale using: **a** 4-by-4 raster and **b** 8-by-8 raster

data analysis. The inferences made from such analyses are either biased, indicating that the observations are spatially aggregated and clustered, or overly precise, indicating that the number of real independent variables is less than the sample size. When the number of real independent variables is less than the sample size, the degree of freedom of the observed data is lower than that assumed in the model.

Scale Dependence of Spatial Autocorrelation

The strength of spatial autocorrelation is often a function of scale or spatial resolution, as illustrated in Fig. 1 using black and white cells. High negative spatial autocorrelation is exhibited in Fig. 1a since each cell has a different color from its neighboring cells. Each cell can be subdivided into four half-size cells (Fig. 1b), assuming the cell's homogeneity. Then, the strength of spatial autocorrelation among the black and white cells increases, while maintaining the same cell arrangement. This illustrates that spatial autocorrelation varies with the study scale.

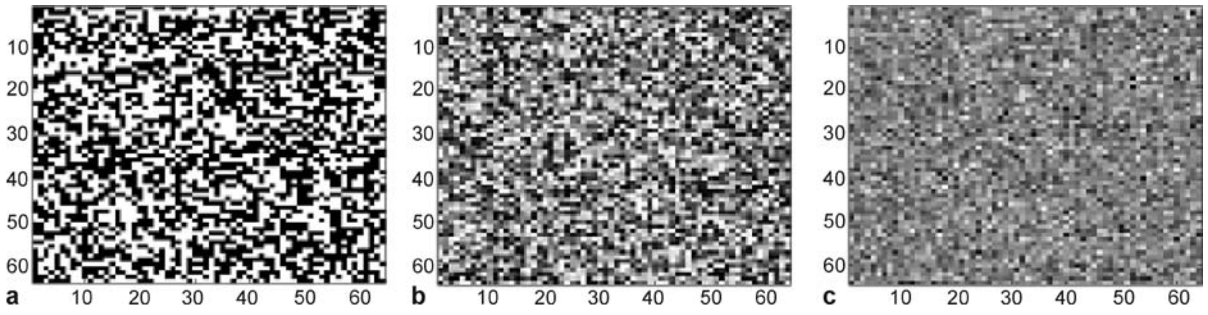
Differentiating Random Data from Spatial Data

Consider three different random distributions and three lattice grids of 64-by-64 cells (see Fig. 2) of each distribution: The first lattice data-set (Fig. 2a) is generated from a binary distribution, the second data-set (Fig. 2b) is generated from a uniform distribution, and the third data-set is generated from a normal distribution. The value at pixel (i, j), $P(i, j)$ $\{P(i, j); i = 1, \ldots, 64, j = 1, \ldots, 64\}$ is assumed to be independent and identically distributed. As shown in Fig. 2, the non-clustering or spatial segregation of the data suggests that the value $P(i, j)$, where $i, j \in R$, has no correlation (zero correlation) with itself in space.

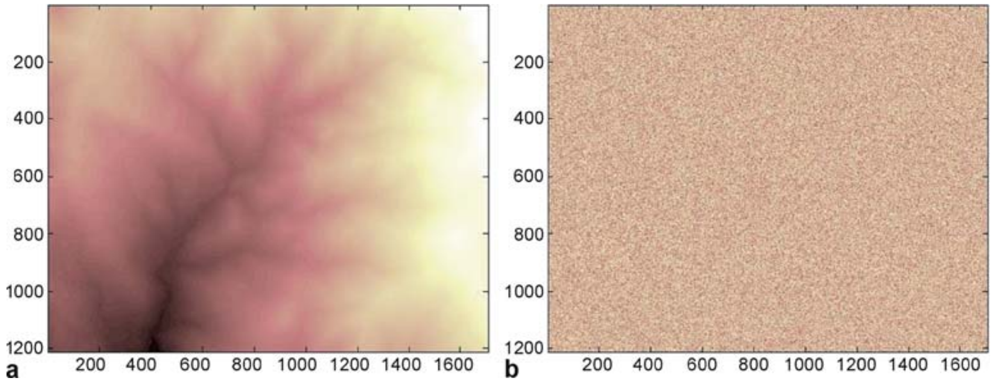
Each pixel (i, j) has eight neighborhoods and each neighborhood also has its own eight adjacent neighborhoods except the cells located on the boundary. The variability of $P(i, j)$ in one direction will not be the same in other directions; thus forming an anisotropic system, indicating the spatial autocorrelation varies in all directions. The quantification of this directional spatial autocorrelation is computationally expensive; thus, the average of each direction at distance k is used to quantify the spatial autocorrelation. The distance k (e. g., k pixel separation of (i, j) in any direction) is called lag distance k. The spatial autocorrelation from each spatial entity to all other entities can be calculated. The average value over all entities of the same lag distance is expressed as a measure of spatial autocorrelation. The above three data-sets are illustrative examples, demonstrating the nonexistence of spatial autocorrelation in randomly generated data-sets.

Consider a digital elevation model (DEM) that shows an array of elevations of the land surface at each spatial location (i, j) as shown in Fig. 3a. The values of this data-set do not change abruptly, whereas in Fig. 3b, the difference of the elevations between the location (i, j) and its neighborhoods changes abruptly as shown in its corresponding color scheme.

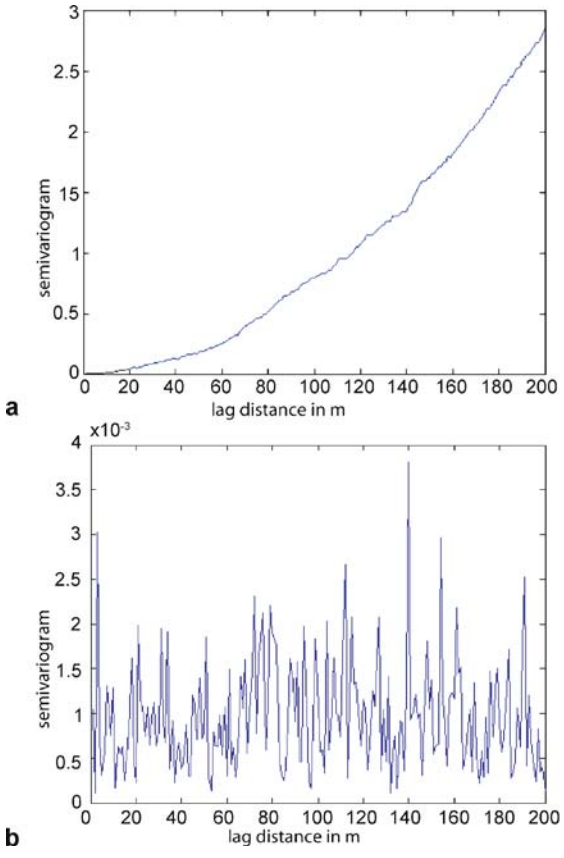
The variogram, a plot of the dissimilarity against the spatial separation (i. e., the lag distance) [12] in spatial data, quantifies spatial autocorrelation and represents how spatial variability changes with lag distance [2]. In Fig. 4a, the semi-variogram value of the DEM surface is zero at the zero lag distance and increases with the lag dis-



Autocorrelation, Spatial, Figure 2 Three different data distributions. **a** Binary distributed data in space. **b** Random uniformly distributed lattice data. **c** Random normally distributed lattice data in space



Autocorrelation, Spatial, Figure 3 **a** One meter spatial resolution LIDAR DEM for south fork Eel, California. **b** One meter normally distributed DEM reconstructed for same statistics (i. e., mean and variance) as LIDAR DEM in **a**



Autocorrelation, Spatial, Figure 4 **a** Variogram for spatial data in Fig. 3a. **b** Variogram for the random data in Fig. 3b

tance, whereas in Fig. 4b, the semi-variogram value of the random surface varies erratically with the increasing lag distance. Contrary to spatial autocorrelation, the semi-variogram has higher values in the absence of spatial correlation and lower values in the presence of spatial correlation. This indicates that spatial autocorrelation gradually disappears as the separation distance increases [10] (Fig. 4a). These variogram figures are generated at a point (x_i) by comparing the values at its four adjacent neighbors such that:

$$\gamma(h) = \frac{1}{N(h)} \sum_{i=1}^{n} (z(x_i) - z(x_i + h))^2 \ , \tag{1}$$

where $z(x_i)$ and $z(x_i + \text{h})$ are the values of the function z located at x_i and $(x_i + \text{h})$, respectively. The four-adjacent-average of the squared difference values along the X and Y axes at lag distance h are used in these variogram clouds. The semi-variogram values in Fig. 4a (generated from lattice data) increase with increasing lag distance whereas the semi-variogram values generated from point data reach a steady state with increasing lag distance.

How to Quantify Spatial Autocorrelation

Several indices can be used to quantify spatial autocorrelation. The most common techniques are Moran's I, Geary's C, and spatial autoregression. These techniques are described in the following sections.

Moran's *I* Method Moran's I index is one of the oldest (Moran, 1950) methods in spatial autocorrelation and is still the de facto standard method of quantifying spatial autocorrelation [8]. This method is applied for points or zones with continuous variables associated with them. The value obtained at a location is compared with the value of other locations. Morgan's I method can be defined as:

$$I = \frac{N \sum_i \sum_j W_{i,j}(X_i - \bar{X})(X_j - \bar{X})}{(\sum_i \sum_j W_{i,j}) \sum_i (X_i - \bar{X})^2} \ , \tag{2}$$

where N is the number of cases, $\bar{X}$ is the mean value of the variable X, X_i and X_j are the values of the variable X at

location i and j, respectively, and $W_{i,j}$ is the weight applied to the comparison between the values at i and j.
The same equation in matrix notation can also be represented as [9]:

$$I = \frac{zWz^t}{zz^t}\,, \tag{3}$$

where $z = (x_1 - \bar{x}, x_2 - \bar{x}, \ldots, x_n - \bar{x})$, z^t is the transpose of matrix z and W is the same contiguity matrix of n- by-n that has been introduced in Eq. 2.
An important property in Moran's I is that the index I depends not only on the variable X, but also on the data's spatial arrangement. The spatial arrangement is quantified by the contiguity matrix, W. If a location i is adjacent to location j, then this spatial arrangement receives the weight of 1; otherwise the value of the weight is 0. Another option is to define W based on the squared inverse distance $(1/d_{ij}^2)$ between the locations i and j [6]. There are also other methods to quantify this contiguity matrix. For example, the sum of the products of the variable x can be compared at locations i and j and then weighted by the inverse distance between i and j.
The value of I is close to 1 or -1 when spatial autocorrelation is high or low, respectively [6].

Geary's C Method Geary's method (Geary, 1954) differs from Moran's method mainly in that the interaction between i and j is measured not as the deviation from the mean, but by the difference of the values of each observation [4]. Geary's C can be defined as:

$$C = \frac{(N-1)\left[\sum_i \sum_j W_{ij}(X_i - X_j)^2\right]}{2\sum_i \sum_j W_{ij}(X_i - \bar{X})^2}\,, \tag{4}$$

where C typically varies between 0 and 2. If the value of one zone is spatially unrelated to any other zone, the expected value of C will be 1. If the value of C is less than 1, a negative spatial autocorrelation is inferred [6]. Geary's C values are inversely related to Moran's I values.
Geary's C and Moran's I will not provide identical inference because the former deals with differences and the latter deals with covariance. The other difference between these two methods is that Moran's I gives a global indication while Geary's C is more sensitive to differences in small neighborhoods [6].

Spatial Autoregression

The disadvantage of linear regression methods is that they assumed iid condition, which is strictly not true for spatial data analysis. Research in spatial statistics has suggested many alternative methods to incorporate spatial dependence into autoregressive regression models, as explained in the following section.

Spatial Autoregressive Regression Model The spatial autoregressive regression model (SAR) is one of the commonly used autoregressive models for spatial data regression. The spatial dependence is introduced into the autoregressive model using the contiguity matrix. Based on this model, the spatial autoregressive regression [9] can be written as:

$$Y = \rho WY + X\beta + \varepsilon\,, \tag{5}$$

where

- Y Observation or dependent variable,
- ρ Spatial autoregressive parameter,
- Y Observation or depend-ant variable,
- W Contiguity matrix,
- β Regressive coefficient,
- α Unobservable error term (N (0,σI)),
- X Feature values or independent variable.

When $\rho = 0$, this model is reduced to the ordinary least square regression equation.
Solution for Eq. 5 is not straightforward and the contiguity matrix W gets quadratic in size compared to the original size of data sample. However, most of the elements of W are zero; thus, sparse matrix techniques are used to speed up the solution process [9].

Illustration of SAR Using Sample Data-Set Consider the following 2-by-2 DEM grid data-set.

100	101
102	103

The contiguity matrix (neighborhood matrix) W can be written as follows:

$$\begin{array}{c} \\ 100 \\ 101 \\ 102 \\ 103 \end{array}\begin{array}{c} \begin{array}{cccc} 100 & 101 & 102 & 103 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

$$\xrightarrow{\text{Normalized contiguity matrix}} \begin{bmatrix} 0 & 0.5 & 0.5 & 0 \\ 0.5 & 0 & 0 & 0.5 \\ 0.5 & 0 & 0 & 0.5 \\ 0 & 0.5 & 0.5 & 0 \end{bmatrix}$$

The normalized contiguity matrix is shown in the right panel. We assumed that $\rho = [0.1]$, $\beta = 1.0 * [1; 2; 3; 4]$, and column vector ε **is** equal to a column vector *(0.01*rand(4,1))*. Then, Eq. 5 can be written as:

$$Y = (QX)\,\beta + \varepsilon\,, \tag{6}$$

where $Q = (I - \rho W^{-1})$.

Demonstration Using Mathworks Matlab Software Matlab software [7] is used to demonstrate this example. The following five matrices are defined for W, ρ, ε, β, and X as $W = [0, 0.5, 0.5, 0; 0.5, 0, 0, 0.5; 0.5, 0, 0, 0.5; 0, 0.5, 0.5, 0]$, $\rho = [0.1]$, $\varepsilon = 0.01*rand(4,1)$, $\beta = 1.0$, $X = [100; 101; 102; 103]$. The above defined values are substituted into Eq. 6, which can be shown in Matlab notation as $y = inv\ (eye(4,4)\text{-}\rho.\ *W)*(\beta\ *X{+}\varepsilon)$. The solution provides an estimation of $y = [111.2874,\ 112.2859,\ 113.2829,\ d\ 114.2786]$.

Key Applications

The key application of spatial autocorrelation is to quantify the spatial dependence of spatial variables. The following are the examples from various disciplines where spatial autocorrelation is used:

Sampling Design

The defined spatial autocorrelation among the contiguous or close locations can be used to answer how large of an area does a single measurement represent. The answer to such questions allows estimates of the best places to make further observations and the number of the samples required in accuracy assessment, and provides useful information for interpolation to estimate values at unobserved locations [13].

Cartography

A main assumption on which statistical estimates of uncertainty are usually based is the independence of the samples during mapping processes. A spatial autocorrelation analysis can be used to test the validity of such an assumption and the related mapping errors. Adjacent elevation differences are usually correlated rather than independent and errors tend to occur in clusters. In addition, the level of accuracy of GIS output products depends on the level of spatial autocorrelation in the source data-sets.

Soil Science

Spatial autocorrelation has been used to study the domain that a soil water content or soil temperature measurement can represent. The distinctive spatial autocorrelations of soil solutes manifests the different reaction and migration patterns for solutes in soil. With a high-resolution soil sampling, a spatial autocorrelation analysis provides another means to delineate boundaries between soil series.

Biology

Patterns and processes of genetic divergence among local populations have been investigated using spatial autocorrelation statistics to describe the autocorrelation of gene frequencies for increasing classes of spatial distance. Spatial autocorrelation analysis has also been used to study a variety of phenomena, such as the genetic structure of plant and animal populations and the distribution of mortality patterns.

Ecology

Ecologists have used spatial autocorrelation statistics to study species–environment relationships. Spatial autocorrelation analysis is a useful tool to investigate mechanisms operating on species richness at different spatial scales [3]. It has shown that spatial autocorrelation can be used to explore how organisms respond to environmental variation at different spatial scales.

Environmental Science

The physical and chemical processes controlling the fate and transport of chemicals in the environment do not operate at random. All measurable environmental parameters exhibit spatial autocorrelation at certain scales [5]. The patterns of spatial autocorrelation in stream water quality can be used to predict water quality impaired stream segments. The spatial autocorrelation test of environmental variables provides important information to the policy-makers for more efficient controls of environmental contaminants.

Risk Assessment

It is often the case that the occurrence of natural hazardous events such as floods and forest fires shows spatial dependence. Spatial autocorrelation allows risk assessment of such undesirable events. It can be used to estimate the probability of a forest fire, as an example, taking place at a specific location. Spatial autocorrelation analysis is also useful in geographical disease clustering tests.

Economics

Because of the heterogeneity across regions and a large number of regions strongly interacting with each other, economic policy measures are targeted at the regional level. Superimposed spatial structures from spatial autocorrelation analysis improve the forecasting performance of non-spatial forecasting models. The spatial dependence and spatial heterogeneity can be used to investigate the effect of income and human capital inequalities on regional economic growth. Spatial autocorrelation analysis is also a useful tool to study the distribution of unemployment rate and price fluctuation within a specific area.

Political Science

After spatial autocorrelation has been defined, geographic units (countries, counties, or census tracts) can be used as predictors of the political outcomes. For example, spatial autocorrelation methods can use geographic data coordinates to check if a location has a significant impact on the voting choice.

Sociology

Spatial autocorrelation has been used to study the correlation between population density and pathology. The spatial interaction has been taken into consideration to study the relationship between population density and fertility. Spatial autocorrelation can also be used to investigate the variations in crime rates and school test scores.

Future Directions

A good knowledge and understanding of spatial autocorrelation is essential in many disciplines which often need predictive inferences from spatial data. Ignoring spatial autocorrelation in spatial data analysis and model development may lead to unreliable and poor fit results. The result of spatial autocorrelation analysis can guide our experiment design, trend analysis, model development, and decision-making. For example, long-term field data monitoring is tedious and costly. Spatial autocorrelation analysis would benefit the design and sampling strategies development of optimal field monitoring sites. Spatial autocorrelation analysis for variables of interest can also assist in the selection of a supermarket location or of a new school.

Cross References

► Semivariogram Modeling

Recommended Reading

1. Moore, D.A., Carpenter, T.E.: Spatial analytical methods and geographic information systems: Use in health research and Epidemiology. Epidemiol. Rev. **21**(2), 143–161 (1999)
2. Devary, J.L., Rice, W.A.: Geostatistis software user's manual for geosciences research and engineering. Pacific Northwest Laboratory, Richland, Washington (1982)
3. Diniz, J.A.F., Bini, L.M., Hawkins, B.A.: Spatial auto-correlation and red herrings in geographical ecology. Glob. Ecol. Biogeogr. **12**, 53–64 (2003)
4. Geary, R.C.: The contiguity ratio and statistical mapping. Inc. Stat. **5**, 115–145 (1954)
5. Haining, R.P.: Spatial data analysis in the social and environmental sciences. Cambridge University Press, Cambridge (1993)
6. Lembo, J.: Lecture 9, Spatial Autocorrelation. http://www.css.cornell.edu/courses/620/lecture9.ppt#256,1. Accessed 10 Apr 2006
7. MATLAB, The Student Edition of MATLAB, The Language of Technical Computing, Version 5, User's Guide. The Mathworks, Inc., New Jersey (1997)
8. Moran, P.A.P.: Notes on continuous stochastic phenomena. Biometrika **37**, 17–23 (1950)
9. Shekhar, S., Chawla, S.: Spatial databases: A tour. Pearson Education Inc., Prentice Hall, Upper Saddle River, NJ (2003)
10. Spatial Autocorrelation, Natural resources Canada. http://www.pfc.forestry.ca/profiles/wulder/mvstats/spatial_e.html. Accessed 10 Apr 2006
11. Tobler, W.R.: A computer movie simulating urban growth in the Detroit region. Econ. Geogr. **46**(2), 234–240 (1970)
12. Wackernagel, H.: Multivariate Geostatistics: An introduction with applications, 3rd edn., pp. 45–58. Springer-Verlag, New York, LLC (2003)
13. Wei, H., Chen, D.: The effect of spatial auto-correlation on the sampling design in accuracy assessment: A case study with simulated data. Environ. Inform. Arch. **2**, 910–919 (2004)
14. Pace, R.K., Barry, R., Sirmans, C.F.: Spatial Statistics and Real Estate. J. Real Estate Finance Econ. **17**, 5–13 (1998)

Automated Map Compilation

► Conflation of Features

Automated Map Generalization

► Feature Extraction, Abstract

Automated Vehicle Location (AVL)

► Intergraph: Real Time Operational Geospatial Applications

Automatic Graphics Generation

► Information Presentation, Dynamic

Automatic Information Extraction

► Data Acquisition, Automation

Autonomy, Space Time

► Time Geography

Autoregressive Models

► Hierarchical Spatial Models

A

Balanced Box Decomposition Tree (Spatial Index)

▶ Nearest Neighbors Problem

Bayesian Estimation

▶ Indoor Positioning, Bayesian Methods

Bayesian Inference

▶ Hurricane Wind Fields, Multivariate Modeling

Bayesian Maximum Entropy

▶ Uncertainty, Modeling with Spatial and Temporal

Bayesian Network Integration with GIS

Daniel P. Ames, Allen Anselmo
Department of Geosciences, Geospatial Software Lab, Idaho State University, Pocatello, ID, USA

Synonyms

Directed acyclic graphs; Probability networks; Influence diagrams; Probabilistic map algebra; Spatial representation of bayesian networks

Definition

A Bayesian Networks (BN) is a graphical-mathematical construct used to probabilistically model processes which include interdependent variables, decisions affecting those variables, and costs associated with the decisions and states of the variables. BNs are inherently system representations and, as such, are often used to model environmental processes. Because of this, there is a natural connection between certain BNs and GIS. BNs are represented as a directed acyclic graph structure with nodes (representing variables, costs, and decisions) and arcs (directed lines representing conditionally probabilistic dependencies between the nodes). A BN can be used for prediction or analysis of real world problems and complex natural systems where statistical correlations can be found between variables or approximated using expert opinion. BNs have a vast array of applications for aiding decision making in areas such as medicine, engineering, natural resources, and decision management. BNs can be used to model geospatially interdependent variables as well as conditional dependencies between geospatial layers. Additionally, BNs have been found to be useful and highly efficient in performing image classification on remotely sensed data.

Historical Background

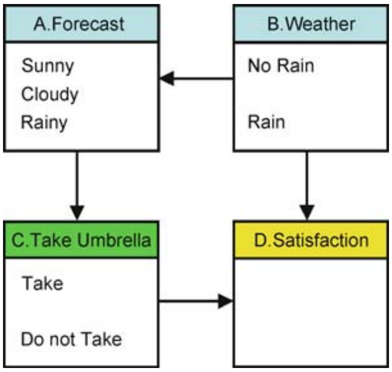
Originally described by Pearl (1988), BNs have been used extensively in medicine and computer science (Heckerman 1997). In recent years, BNs have been applied in spatially explicit environmental management studies. Examples include: the Neuse Estuary Bayesian ecological response network (Borsuk and Reckhow 2000), Baltic salmon management (Varis and Kuikka 1996), climate change impacts on Finnish watersheds (Kuikka and Varis 1997), the Interior Columbia Basin Ecosystem Management Project (Lee and Bradshaw 1998), and waterbody eutrophication (Haas 1998). As illustrated in these studies, a BN graph structures a problem such that it is visually interpretable by stakeholders and decision-makers while, serving as an efficient means for evaluating the probable outcomes of management decisions on selected variables.

Both BNs and GIS can be used to represent spatially explicit, probabilistically connected environmental and other systems, however the integration of the two techniques has only been explored relatively recently. BN integration with GIS typically takes one of four distinct forms: 1) BN-based layer combination (i. e. probabilistic map-algebra) as demonstrated in Taylor (2003); 2) BN-based classification as demonstrated in Stassopoulou et al. (1998) and Stassopoulou and Caelli (2000); 3) Using BNs for intelligent, spatially-oriented data retrieval, as demonstrated in Walker et al. (2004) and Walker et al. (2005); and 4) GIS-based BN decision support system (DSS) frameworks where BN nodes are spatially represented in a GIS framework as presented by Ames (2005).

Scientific Fundamentals

As noted above, BNs are used to model reality by representing conditional probabilistic dependencies between interdependent variables, decisions, and outcomes. This section provides an in-depth explanation of BN analysis using an example BN model called the "Umbrella" BN (Fig. 1), an augmented version of the well-known "Weather" influence diagram presented by Shachter and Peot (1992). This simple BN attempts to model the variables and outcomes associated with the decision to take or not take an umbrella on a given outing. This problem is represented in the BN by four nodes. "Weather" and "Forecast" are nature or chance nodes where "Forecast" is conditioned on the state of "Weather" and "Weather" is treated as a random variable with a prior probability distribution based on historical conditions. "Take Umbrella" is a decision variable that, together with the "Weather" variable defines the status of "Satisfaction". The "Satisfaction" node is known as a "utility" or "value" node. This node associates a resultant outcome value (monetary or otherwise) to represent the satisfaction of the individual based on the decision to take the umbrella and whether or not there is rain. Each of these BN nodes contains discrete states where each variable state represents abstract events, conditions, or numeric ranges of each variable.

The Umbrella model can be interpreted as follows: if it is raining, there is a higher probability that the forecast will predict it will rain. In reverse, through the Bayesian network "backward propagation of evidence" if the forecast predicts rain it can be inferred that there is a higher chance that rain will actually occur. The link between "Forecast" and "Take Umbrella" indicates that the "Take Umbrella" decision is based largely on the observed forecast. Finally, the link to the "Satisfaction" utility node from both "Take Umbrella" and "Weather" captures the relative gains in satisfaction derived from every combination of states of the BN variables.



Bayesian Network Integration with GIS, Figure 1 Umbrella Bayesian Decision Network Structure. *A* and *B* nature nodes, *C* a decision node, and *D* a utility node

Bayesian networks are governed by two mathematical techniques: conditional probability and Bayes' theorem.

Conditional probability is defined as the probability of one event given the occurrence of another event and can be calculated as the joint probability of the two events occurring divided by the probability of the second event:

$$P(A|B) = \frac{P(A, B)}{P(B)} . \tag{1}$$

From Eq. 1, the fundamental rule for probability calculus and the downward propagation of evidence in a BN can be derived. Specifically, it is seen that the joint probability of A and B equals the conditional probability of event A given B, multiplied by the probability of event B (Eq. 2).

$$P(A, B) = P(A|B) \cdot P(B) . \tag{2}$$

Equation 2 is used to compute the probability of any state in the Bayesian network given the states of the parent node events. In Eq. 3, the probability of state A_x occurring given parent B is the sum of the probabilities of the state of A_x given state B_i, with i being an index to the states of B, multiplied by the probability of that state of B.

$$P(A_x, B) = \sum_i P(A_x|B_i) \cdot P(B_i) . \tag{3}$$

Similarly, for calculating states with multiple parent nodes, the equation is modified to make the summation of the conditional probability of the state A_x given states B_i and C_j multiplied by the individual probabilities of B_i and C_j.

$$P(A_x, B, C) = \sum_{i,j} P(A_x|B_i, C_j) \cdot P(B_i) \cdot P(C_j) . \tag{4}$$

Finally, though similar in form, utility nodes do not calculate probability, but instead calculate the utility value as a metric or index given the states of its parent or parents as shown in Eqs. 5 and 6.

$$U(A, B) = \sum_i U(A|B_i) \cdot P(B_i) \tag{5}$$

$$U(A, B, C) = \sum_{i,j} U(A|B_i, C_j) \cdot P(B_i) \cdot P(C_j) . \tag{6}$$

The second equation that is critical to BN modeling is Bayes' theorem:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \,. \tag{7}$$

The conditional probability inversion represented here allows for the powerful technique of Bayesian inference, for which BNs are particularly well suited. In the Umbrella model, inferring a higher probability of a rain given a rainy forecast is an example application of Bayes' Theorem.

Connecting each node in the BN is a conditional probability table (CPT). Each nature node (state variable) includes a CPT that stores the probability distribution for the possible states of the variable given every combination of the states of its parent nodes (if any). These probability distributions can be assigned by frequency analysis of the variables, expert opinion based on observation or experience, or they can be set to some "prior" distribution based on observations of equivalent systems.

Tables 1 and 2 show CPTs for the Umbrella BN. In Table 1, the probability distribution of rain is represented as 70% chance of no rain, and 30% chance of rain. This CPT can be assumed to be derived from historical observations of the frequency of rain in the given locale. Table 2, represents the probability distribution of the possible weather forecasts ("Sunny", "Cloudy", or "Rainy") conditioned on the actual weather event. For example, when actually rained, the prior forecast called for "Rainy" 60% of the time, "Cloudy" 25% of the time, and "Sunny" 15% of the time. Again, these probabilities can be derived from historical observations of prediction accuracies or from expert judgment.

Table 3 is a utility table defining the relative gains in utility (in terms of generic "units of satisfaction) under all of the possible states of the BN. Here, satisfaction is highest when there is no rain and the umbrella is not taken and lowest when the umbrella is not taken but it does rain. Satisfaction "units" are in this case assigned as arbitrary ratings from 0 to 100, but in more complex systems, utility can be used to represent monetary or other measures.

Following is a brief explanation of the implementation and use of the Umbrella BN. First it is useful to compute $P(\text{Forecast} = \text{Sunny})$ given unknown Weather conditions as follows:

$$\begin{aligned} &P(\text{Forecast} = \text{Sunny}) \\ &\quad = \sum_{i=\{\text{NoRain,Rain}\}} P(\text{Forecast} = \text{Sunny}|\text{Weather}_i) \cdot P(\text{Weather}_i) \\ &\quad = 0.7 \cdot 0.7 + 0.15 \cdot 0.3 = 0.535 = 54\% \,. \end{aligned}$$

Next $P(\text{Forecast} = \text{Cloudy})$ and $P(\text{Forecast} = \text{Rainy})$ can be computed as:

$$\begin{aligned} &P(\text{Forecast} = \text{Cloudy, Weather}) \\ &\quad = 0.2 \cdot 0.7 + 0.25 \cdot 0.3 = 0.215 = 22\% \\ &P(\text{Forecast} = \text{Cloudy, Weather}) \\ &\quad = 0.1 \cdot 0.7 + 0.6 \cdot 0.3 = 0.25 = 25\% \,. \end{aligned}$$

Finally, evaluate the "Satisfaction" utility under both possible decision scenarios (take or leave the umbrella).

$$\begin{aligned} &U(\text{Satisfaction}|\text{TakeUmbrella} = \text{Take}) \\ &= \sum_{i,j} U(\text{Satisfaction}|\text{TakeUmbrella, Weather}_j) \\ &\qquad \cdot P(\text{TakeUmbreall}a_i) \cdot P(\text{Weather}_j) \\ &= 20 * 1.0 * 0.7 + 100 * 0.0 * 0.7 + 70 * 1.0 * 0.3 \\ &\quad + 0 * 0.0 * 0.3 = 35 \,. \end{aligned}$$

Similarly, the utility of not taking the umbrella is computed as:

$$\begin{aligned} &U(\text{Satisfaction, TakeUmbrella} = \text{NoTake, Weather}) \\ &= 20 * 0.0 * 0.7 + 100 * 1.0 * 0.7 + 70 * 0.0 * 0.3 \\ &\qquad + 0 * 1.0 * 0.3 = 70 \end{aligned}$$

Clearly, the higher satisfaction is predicted for leaving the umbrella at home, thereby providing an example of how a simple BN analysis can aid the decision-making process. While the Umbrella BN presented here is quite simple and not particularly spatially explicit, it serves as a generic BN

Bayesian Network Integration with GIS, Table 1 Probability of rain

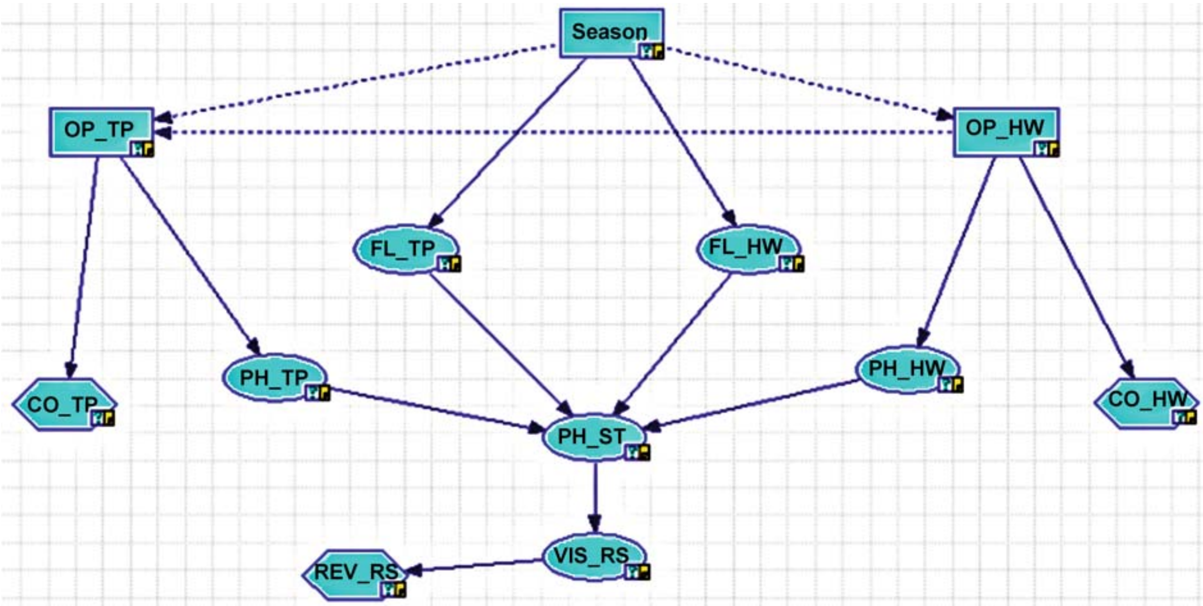
Weather	
No Rain	Rain
70%	30%

Bayesian Network Integration with GIS, Table 2 Forecast probability conditioned on rain

	Forcast		
Weather	Sunny	Cloudy	Rainy
No Rain	70%	20%	10%
Rain	15%	25%	60%

Bayesian Network Integration with GIS, Table 3 Satisfaction utility conditioned on rain and the "Take Umbrella" decision

	Satisfaction	
Weather	Take Umbrella	Satisfaction
No Rain	Take	20 units
No Rain	Do not Take	100 units
Rain	Take	70 units
Rain	Do not Take	0 units



Bayesian Network Integration with GIS, Figure 2 The East Canyon Creek BDN from Ames et al. (2005), as seen in the GeNle (Decision Systems Laboratory 2006) graphical node editor application

example. Specific application of BNs in GIS is presented in the following section.

Key Applications

As discussed before, integration of GIS and BNs is useful in any BN which has spatial components, whether displaying a spatially-oriented BN, using GIS functionality as input to a BN, or forming a BN from GIS analysis. Given this, the applications of such integration are only limited by that spatial association really. One example mentioned above of such a spatial orientation was showed usefulness of a watershed management BN, but there are other types of BNs which may benefit from this form of integration. For instance, many ecological, sociological, and geological studies which might benefit from a BN also could have strong spatial associations. Another example might be that traffic analyses BNs have very clear spatial associations often. Finally, even BNs trying to characterize the spread of diseases in epidemiology would likely have clear spatial association.

As outlined above, GIS-based BN analysis typically takes one of four distinct forms including:

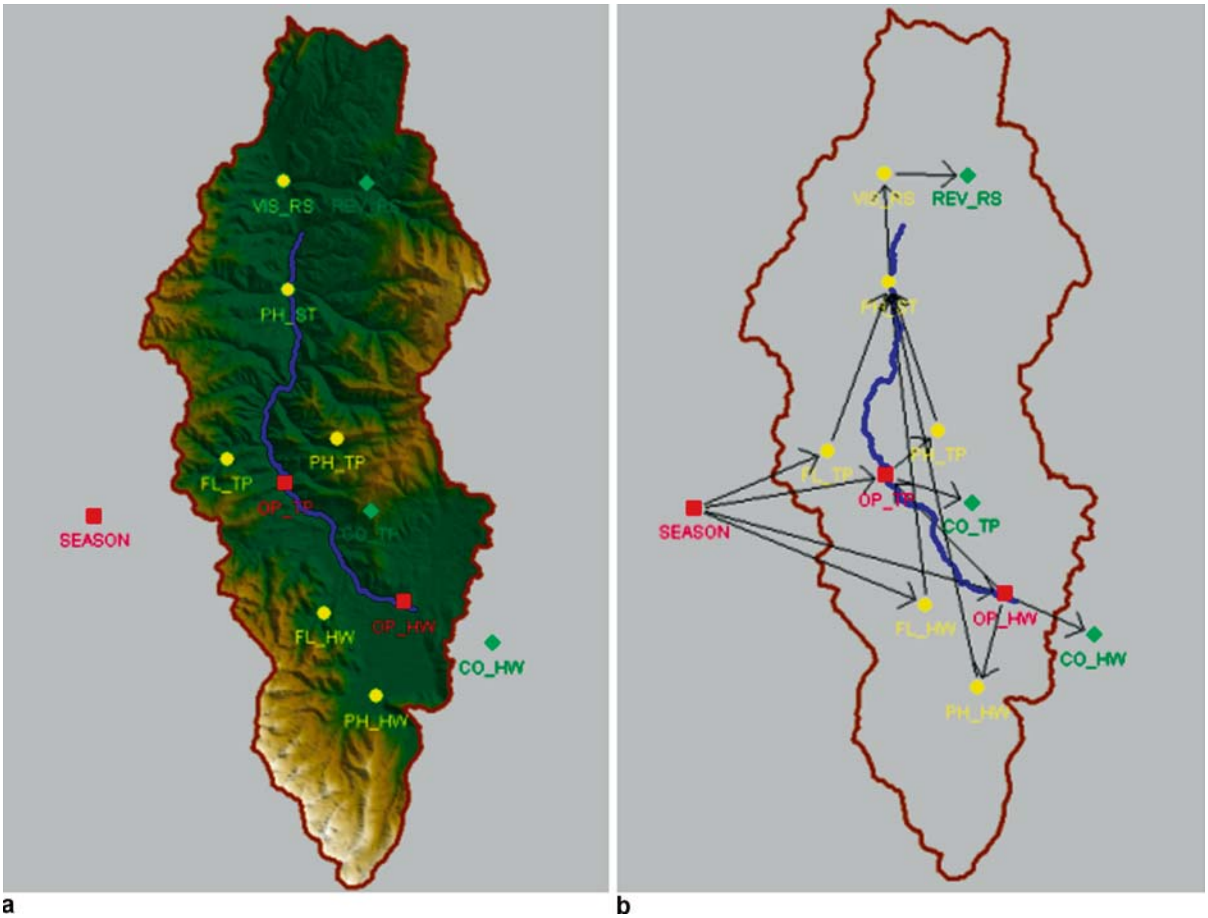
- Probabilistic map-algebra
- Image classification
- Automated data query and retrieval
- Spatial representation of BN nodes

A brief explanation of the scientific fundamentals of each of these uses is presented here.

Probabilistic Map Algebra

Probabilistic map algebra involves the use of a BN as the combinatorial function used on a cell by cell basis when combining raster layers. For example, consider the ecological habitat models described by Taylor (2003). Here, several geospatial raster data sets are derived representing proximity zones for human caused landscape disturbances associated with development of roads, wells, and pipelines. Additional data layers representing known habitat for each of several threatened and endangered species are also developed and overlaid on the disturbance layers. Next a BN was constructed representing the probability of habitat risk conditioned on the both human disturbance and habitat locations. CPTs in this BN were derived from interviews with acknowledged ecological experts in the region. Finally, this BN was applied on a cell by cell basis throughout the study area, resulting in a risk probability map for the region for each species of interest.

Use of BNs in this kind of probabilistic map algebra is currently hindered only by the lack of specialized tools to support the analysis. However, the concept holds significant promise as an alternative to the more traditional GIS based "indicator analysis" where each layer is reclassified to represent an arbitrary index and then summed to give a final metric (often on a 1 to 100 scale of either suitability or unsuitability). Indeed the BN approach results in a more interpretable probability map. For example, such an analysis could be used to generate a map of the probabil-



Bayesian Network Integration with GIS, Figure 3 **a** East Canyon displayed with the East Canyon BN overlain on it. **b** Same, but with the DEM layer turned off and the BN network lines displayed

ity of landslide conditioned on slope, wetness, vegetation etc. Certainly a map that indicates percent chance of landslide could be more informative for decision makers, than an indicator model that simply displays the sum of some number of reclassified indicators.

Image Classification

In the previous examples, BN CPTs are derived from historical data or information from experts. However, many BN applications make use of the concept of Bayesian learning as a means of automatically estimating probabilities from existing data. BN learning involves a formal automated process of "creating" and "pruning" the BN node-arc structure based on rules intended to maximize the amount of unique information represented by the BN CPTs. In a GIS context, BN learning algorithms have been extensively applied to image classification problems. Image classification using a BN requires the identification of a set of input layers (typically multispectral or hyperspectral bands) from which a known set of objects or classifications are to be identified.

Learning datasets include both input and output layers where output layers clearly indicate features of the required classes (e. g. polygons indicating known land cover types). A BN learning algorithm applied to such a data set will produce a optimal (in BN terms) model for predicting land cover or other classification scheme at a given raster cell based on the input layers. The application of the final BN model to predict land cover or other classification at an unknown point is similar to the probabilistic map algebra described previously.

Automated Data Query and Retrieval

In the case of application of BNs to automated query and retrieval of geospatial data sets, the goal is typically to use expert knowledge to define the CPTs that govern which data layers are loaded for visualization and analysis. Using this approach in a dynamic web-based mapping system, one could develop a BN for the display of layers using a CPT that indicates the probability that the layer is important, given the presence or absence of other layers or features within layers at the current view extents. Such a tool would supplant the typical approach which is to activate or deactivate layers based strictly on "zoom level". For example, consider a military GIS mapping system used to identify proposed targets. A BN-based data retrieval system could significantly optimize data transfer and bandwidth usage by only showing specific high resolution imagery when the probability of needing that data is raised due to the presence of other features which indicate a higher likelihood of the presence of the specific target.

BN-based data query and retrieval systems can also benefit from Bayesian learning capabilities by updating CPTs with new information or evidence observed during the use of the BN. For example, if a user continually views several datasets simultaneously at a particular zoom level or in a specific zone, this increases the probability that those data sets are interrelated and will should result in modified CPTs representing those conditional relationships.

Spatial Representation of BN Nodes

Many BN problems and analyses though not completely based on geospatial data have a clear geospatial component and as such can be mapped on the landscape. This combined BN-GIS methodology is relatively new but has significant potential for helping improve the use and understanding of a BN. For example, consider the East Canyon Creek BN (Ames et al. 2005) represented in Fig. 2. This BN is a model of streamflow (FL_TP and FL_HW) at both a wastewater treatment plant and in the stream headwaters, conditional on the current season (SEASON). Also the model includes estimates of phosphorus concentrations at the treatment plant and in the headwaters (PH_TP and PH_HW) conditional on season and also on operations at both the treatment plant (OP_TP) and in the headwaters (OP_HW). Each of these variables affect phosphorus concentrations in the stream (PH_ST) and ultimately reservoir visitation (VIS_RS). Costs of operations (CO_TP and CO_HW) as well as revenue at the reservoir (REV_RS) are represented as utility nodes in the BN.

Most of the nodes in this BN (except for SEASON) have an explicit spatial location (i. e. they represent conditions at a specific place). Because of this intrinsic spatiality, the East Canyon BN can be represented in a GIS with points indicating nodes and arrows indicating the BN arcs (i. e. Fig. 3). Such a representation of a BN within a GIS can give the end users a greater understanding of the context and meaning of the BN nodes. Additionally, in many cases, it may be that the BN nodes correspond to specific geospatial features (e. g. a particular weather station) in which case spatial representation of the BN nodes in a GIS can be particularly meaningful.

Future Directions

It is expected that research and development of tools for the combined integration of GIS and BNs will continue in both academia and commercial entities. New advancements in each of the application areas described are occurring on a regular basis, and represent and active and interesting study area for many GIS analysts and users.

Recommended Reading

1. Ames, D.P.: Bayesian Decision Networks for Watershed Management. Utah State University, Logan, UT (2002)
2. Ames, D.P., Neilson, B.T., Stevens, D.K., Lall, U.: Using Bayesian networks to model watershed management decisions: an East Canyon Creek case study. IWA Publishing, J. Hydroinform. (2005)
3. Borsuk, M.E., Reckhow, K.H.: Summary Description of the Neuse Estuary Bayesian Ecological Response Network (Neu-BERN). (2000) http://www2.ncsu.edu/ncsu/CIL/WRRI/neuseltm.html (Dec. 26, 2001)
4. Haas, T.C.: Modeling waterbody eutrophication with a Bayesian belief network. School of Business Administration, Working Paper. University of Wisconsin, Milwaukee, Wisconsin (1998)
5. Heckerman, D.: Bayesian networks for data mining. Data Mining and Knowledge Discovery, **1**, 79–119. MapWindow Open Source Team (2007). MapWindow GIS 4.3 Open Source Software, Accessed February 06, 2007 at the MapWindow Website: http://www.mapwindow.org/ (1997)
6. Kuikka, S., Varis, O.: Uncertainties of climate change impacts in Finnish watersheds: a Bayesian network analysis of expert knowledge. Boreal Environ. Res. **2**, 109–128 (1997)
7. Lee, D.C., Bradshaw, G.A.: Making Monitoring Work for Managers: Thoughts on a conceptual framework for improved monitoring within broad-scale ecosystem management. (1998) http://icebmp.gov/spatial/lee_monitor/preface.html (Dec. 26, 2001)
8. Norsys Software Corp.: Netica Bayesian Belief Network Software. Acquired from http://www.norsys.com/ (2006)
9. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Francisco (1988)
10. Stassopoulou, A., Petrou, M., Kittler, J.: Application of a Bayesian network in a GIS based decision making system. Int. J. Geograph. Info. Sci. **12**(1) 23–45 (1998)
11. Stassopoulou, A., Caelli, T.: Building detection using Bayesian networks. Int. J. Pattern Recognit. Artif. Intell. **14**(6), 715–733 (2000)
12. Taylor, K.J.: Bayesian Belief Networks: A Conceptual Approach to Assessing Risk to Habitat, Utah State University, Logan, UT (2003)
13. Varis, O., Kuikka, S.: An influence diagram approach to Baltic salmon management. Proc. Conference on Decision Analysis for Public Policy in Europe, INFORMS Decision Analysis Society, Atlanta, GA (1996)
14. Walker, A., Pham, B., Maeder, A.: A Bayesian framework for automated dataset retrieval. In: Geographic Information Systems,

mmm, p. 138. 10th International Multimedia Modelling Conference (2004)
15. Walker, A., Pham, B., Moody, M.: Spatial Bayesian learning algorithms for geographic information retrieval. In: Proceedings 13th annual ACM international workshop on Geographic information systems, pp. 105–114. Bremen, Germany (2005)
16. Shachter, R., Peot, M.: Decision making using probabilistic inference methods. In: Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence, pp. 275–283 (1992)

Bayesian Spatial Regression

► Bayesian Spatial Regression for Multi-source Predictive Mapping

Bayesian Spatial Regression for Multi-source Predictive Mapping

ANDREW O. FINLEY[1], SUDIPTO BANERJEE[2]
[1] Department of Forestry and Department of Geography, Michigan State University, East Lansing, MI, USA
[2] Division of Biostatistics, School of Public Health, University of Minnesota, Mineapolis, MN, USA

Synonyms

Pixel-based prediction; Bayesian spatial regression; Spatial regression

Definition

Georeferenced ground measurements for attributes of interest and a host of remotely sensed variables are coupled within a Bayesian spatial regression model to provide predictions across the domain of interest. As the name suggests, multi-source refers to multiple sources of data which share a common coordinate system and can be linked to form sets of regressands or response variables, $y(\mathbf{s})$, and regressors or covariates, $\mathbf{x}(\mathbf{s})$, where the $\mathbf{s}$ denotes a known location in $\mathbb{R}^2$ (e. g., easting–northing or latitude–longitude). Interest here is in producing spatially explicit predictions of the response variables using the set of covariates. Typically, the covariates can be measured at any location across the domain of interest and help explain the variation in the set of response variables. Within a multi-source setting, covariates commonly include multitemporal spectral components from remotely sensed images, topographic variables (e. g., elevation, slope, aspect) from a digital elevation model (DEM), and variables derived from vector or raster maps (e. g., current or historic land use, distance to stream or road, soil type, etc.). Numerous methods have been used to map the set of response variables. The focus here is linking the $y(\mathbf{s})$ and $\mathbf{x}(\mathbf{s})$ through Bayesian spatial regression models. These models provide unmatched flexibility for partitioning sources of variability (e. g., spatial, temporal, random), simultaneously predicting multiple response variables (i. e., multivariate or vector spatial regression), and providing access to the full posterior predictive distribution of any base map unit (e. g., pixel, mulitpixel, or polygon). This chapter offers a brief overview of remotely sensed data which is followed by a more in-depth presentation of Bayesian spatial modeling for multi-source predictive mapping. Multi-source forest inventory data is used to illustrate aspects of the modeling process.

Historical Background

In 1970, the National Academy of Sciences recognized remote sensing as "the joint effects of employing modern sensors, data-processing equipment, information theory and processing methodology, communications theory and devices, space and airborne vehicles, and large-systems theory and practices for the purpose of carrying out aerial or space surveys of the earth's surface" [26] p1. In the nearly four decades since this definition was offered, every topic noted has enjoyed productive research and development. As a result, a diverse set of disciplines routinely use remotely sensed data including: natural resource management; hazard assessment; environmental assessment; precision farming and agricultural yield assessment; coastal and oceanic monitoring; fresh water quality assessment; and public health. Several key publications document these advancements in remote sensing research and application including *Remote Sensing of Environment*, *Photogrammetric Engineering and Remote Sensing*, *International Journal of Remote Sensing*, and *IEEE Transactions on Geoscience and Remote Sensing*.

With the emergence of highly efficient Geographical Information Systems (GIS) databases and associated software, the modeling and analysis of spatially referenced data sets have also received much attention over the last decade. In parallel with the use of remotely sensed data, spatially-referenced data sets and their analysis using GIS is often an integral part of scientific and engineering investigations; see, for example, texts in geological and environmental sciences [38], ecological systems [33], digital terrain cartography [21], computer experiments [31], and public health [11]. The last decade has also seen significant development in statistical modeling of complex spatial data; see, for example, the texts by [9,10,24,32,36] for a variety of methods and applications.

A new approach that has recently garnered popularity in spatial modeling follows the Bayesian inferential paradigm. Here, one constructs hierarchical (or multi-level) schemes by assigning probability distributions to parameters *a priori* and inference is based upon the distribution of the parameters conditional upon the data *a posteriori*. By modeling both the observed data and any unknown regressor or covariate effects as random variables, the hierarchical Bayesian approach to statistical analysis provides a cohesive framework for combining complex data models and external knowledge or expert opinion. A theoretical foundation for contemporary Bayesian modeling can be found in several key texts, including [2,7,16,30].

Scientific Fundamentals

This chapter focuses on predictive models that use covariates derived from digital imagery captured by sensors mounted on orbiting satellites. These modern spaceborne sensors are categorized as either passive or active. Passive sensors detect the reflected or emitted electromagnetic radiation from natural sources (typically solar energy), while active sensors emit energy that travels to the surface feature and is reflected back toward the sensor, such as radar or LIDAR (Light Detection and Ranging). The discussion and illustration covered here focus on data from passive sensors, but can be extended to imagery obtained from active sensors.

The resolution and scale are additional sensor characteristics. There are three components to resolution: 1) spatial resolution refers to the size of the image pixel, with high spatial resolution corresponding to small pixel size; 2) radiometric resolution is the sensor's ability to resolve levels of brightness, a sensor with high radiometric resolution can distinguish between many levels of brightness; 3) spectral resolution describes the sensor's ability to define wavelength intervals, a sensor with high spectral resolution can record many narrow wavelength intervals. These three components are related. Specifically, higher spatial resolution (i. e., smaller pixel size) results in lower radiometric and/or spectral resolution. In general terms, if pixel size is large, the sensor receives a more robust signal and can then distinguish between a smaller degree of radiometric and spectral change. As the pixel size decreases the signal is reduced and so too is the sensor's ability to detect changes in brightness. Scale refers to the geographic extent of the image or scene recorded by the sensor. Scale and spatial resolution hold an inverse relationship; that is, the greater the spatial resolution, the smaller the extent of the image.

In addition to the academic publications noted above, numerous texts (see e. g., [8,25,29]) provide detail on acquiring and processing remotely sensed imagery for use in prediction models. The modeling illustrations offered in this chapter, use imagery acquired from the Thematic Mapper (TM) and Enhanced Thematic Mapper Plus (ETM+) sensors mounted on the Landsat 5 and Landsat 7 satellites, respectively (see e. g., http://landsat.gsfc.nasa.gov for more details). These are considered mid-resolution sensors because the imagery has moderate spatial, radiometric, and spectral resolution. Specifically, the sensors record reflected or emitted radiation in blue-green (band 1), green (band 2) red (band 3), near-infrared (band 4), mid-infrared (bands 5 and 7), and far-infrared (band 6) portions of the electromagnetic spectrum. Their radiometric resolution within the bands record brightness at 265 levels (i. e., 8 bits) with a spatial resolution of 30×30 meter pixels (with the exception of band 6 which is 120×120). The scale of these images is typically 185 km wide by 170 km long, which is ideal for large area moderate resolution mapping.

In addition to the remotely sensed covariates, predictive models require georeferenced measurements of the response variables of interest. Two base units of measure and mapping are commonly encountered: locations that are areas or regions with well-defined neighbors (such as pixels in a lattice, counties in a map, etc.), whence they are called *areally referenced* data; or locations that are points with coordinates (latitude-longitude, easting-northing, etc.), in which case they are called *point referenced* or *geostatistical*. Statistical theory and methods play a crucial role in the modeling and analysis of such data by developing spatial process models, also known as stochastic process or random function models, that help in predicting and estimating physical phenomena. This chapter deals with the latter – modeling of point-referenced data.

The methods and accompanying illustration presented here provide pixel-level prediction at the lowest spatial resolution offered in the set of remotely sensed covariates. In the simplest setting it is assumed that the remotely sensed covariates cover the entire area of interest, referred to as the domain, $\mathcal{D}$. Further, all covariates share a common spatial resolution (not necessarily common radiometric or spectral resolution). Finally, each point-referenced location $\mathbf{s}$, in the set $S = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$, where a response variable is measured must coincide with a covariate pixel. In this way, the elements in the $n \times 1$ response vector, $\mathbf{y} = [y(\mathbf{s}_i)]_{i=1}^n$, are uniquely associated with the rows of the $n \times p$ covariate matrix, $X = [\mathbf{x}^\mathrm{T}(\mathbf{s}_i)]_{i=1}^n$. This statement suggest that given the N pixels which define $\mathcal{D}$, n of them are associated with a known response value and $n^* = N - n$ require prediction. This is the typical set-up for model-based predictive mapping.

The univariate spatial regression model for point-referenced data is written as

$$y(\mathbf{s}) = \mathbf{x}^\mathrm{T}(\mathbf{s})\boldsymbol{\beta} + w(\mathbf{s}) + \epsilon(\mathbf{s}) \,, \tag{1}$$

where $\{w(\mathbf{s}) : \mathbf{s} \in \mathcal{D}\}$ is a *spatial random field*, with $\mathcal{D}$ an open subset of $\mathbb{R}^d$ of dimension d; in most practical settings $d = 2$ or $d = 3$. A random field is said to be a *valid* spatial process if for any finite collection of sites S of arbitrary size, the vector $\mathbf{w} = [w(s_i)]_{i=1}^n$ follows a well-defined joint probability distribution. Also, $\epsilon(\mathbf{s}) \overset{iid}{\sim} N(0, \tau^2)$ is a *white-noise* process, often called the *nugget* effect, modeling measurement error or micro-scale variation (see, e. g., [9]).

A popular modeling choice for a spatial random field is the *Gaussian process*, $w(\mathbf{s}) \sim GP(0, K(\cdot,\cdot))$, specified by a valid covariance function $K(\mathbf{s}, \mathbf{s}'; \boldsymbol{\theta}) = Cov(w(\mathbf{s}), w(\mathbf{s}'))$ that models the covariance corresponding to a pair of sites $\mathbf{s}$ and $\mathbf{s}'$. This specifies the joint distribution for $\mathbf{w}$ as $MVN(\mathbf{0}, \Sigma_\mathbf{w})$, where $\Sigma_\mathbf{w} = [K(\mathbf{s}_i, \mathbf{s}_j; \boldsymbol{\theta})]_{i,j=1}^n$ is the $n \times n$ covariance matrix with *(i,j)*-th element given by $K(\mathbf{s}_i, \mathbf{s}_j; \boldsymbol{\theta})$. Clearly $K(\mathbf{s}, \mathbf{s}'; \boldsymbol{\theta})$ cannot be just any function; it must ensure that the resulting $\Sigma_\mathbf{w}$ matrix is symmetric and positive definite. Such functions are known as positive definite functions and are characterized as the characteristic function of a symmetric random variable (due to a famous theorem due to Bochner). Further technical details about positive definite functions can be found in [2,9,10].

For valid inference on model parameters and subsequent prediction model (1) requires that the underlying spatial random field be *stationary* and *isotropic*. Stationarity, in spatial modeling contexts, refers to the setting when $\mathbf{K}(\mathbf{s}, \mathbf{s}'; \boldsymbol{\theta}) = \mathbf{K}(\mathbf{s} - \mathbf{s}'; \boldsymbol{\theta})$; that is, the covariance function depends upon the separation of the sites. Isotropy goes further and specifies $\mathbf{K}(\mathbf{s}, \mathbf{s}') = \sigma^2 \rho(\mathbf{s}, \mathbf{s}'; \boldsymbol{\theta})$, where $\|\mathbf{s} - \mathbf{s}'\|$ is the distance between the sites. Usually one further specifies $K(\mathbf{s}, \mathbf{s}'; \boldsymbol{\theta}) = \sigma^2 \rho(\mathbf{s}, \mathbf{s}'; \boldsymbol{\theta})$ where $\rho(\cdot; \boldsymbol{\theta})$ is a *correlation function* and $\boldsymbol{\theta}$ includes parameters quantifying rate of correlation decay and smoothness of the surface $w(\mathbf{s})$. Then $Var(w(\mathbf{s})) = \sigma^2$ represents a spatial variance component in the model in (1). A very versatile class of correlation functions is the Matérn correlation function given by

$$\begin{aligned} \rho\left(\|\mathbf{s} - \mathbf{s}'\|; \boldsymbol{\theta}\right) &= \frac{1}{2^{\nu-1}\Gamma(\nu)} \left(\|\mathbf{s} - \mathbf{s}'\| \phi\right)^\nu \\ &\quad \cdot \mathcal{K}_\nu\left(\|\mathbf{s} - \mathbf{s}'\|; \phi\right); \quad \phi > 0,\ \nu > 0, \end{aligned} \tag{2}$$

where $\boldsymbol{\theta} = (\phi, \nu)$ with ϕ controlling the decay in spatial correlation and ν yielding smoother process realizations for higher values. Also, Γ is the usual Gamma function while $\mathcal{K}_\nu$ is a modified Bessel function of the third kind with order ν and $\|\mathbf{s} - \mathbf{s}'\|$ is the Euclidean distance between the sites $\mathbf{s}$ and $\mathbf{s}'$.

With observations $\mathbf{y}$ from n locations, the data likelihood is written in the marginalized form $\mathbf{y} \sim MVN(\mathbf{X}\boldsymbol{\beta}, \Sigma_\mathbf{y})$, with $\Sigma_\mathbf{y} = \sigma^2 R(\boldsymbol{\theta}) + \tau^2 I_n$ and $R(\theta) = [\rho(\mathbf{s}_i, \mathbf{s}_j; \boldsymbol{\theta})]_{i,j=1}^n$ is the spatial correlation matrix corresponding to $w(\mathbf{s})$. For hierarchical models, one assigns prior (hyperprior) distributions to the model parameters (hyperparameters) and inference proceeds by sampling from the posterior distribution of the parameters (see, e. g., [2]). Generically denoting by $\Omega = (\boldsymbol{\beta}, \tau^2, \theta, \sigma^2)$ the set of parameters that are to be updated in the marginalized model from, sample from the posterior distribution

$$p(\Omega \mid \mathbf{y}) \propto p(\boldsymbol{\beta}) p(\tau^2) p(\boldsymbol{\theta}) p(\sigma^2) p(\mathbf{y} \mid \boldsymbol{\beta}, \tau^2, \theta, \sigma^2)\,. \tag{3}$$

An efficient Markov Chain Monte Carlo (MCMC) algorithm is obtained by updating $\boldsymbol{\beta}$ from its full conditional $MVN(\boldsymbol{\mu}_{\boldsymbol{\beta}|\cdot}, \Sigma_{\boldsymbol{\beta}|\cdot})$, where

$$\begin{aligned} \Sigma_{\boldsymbol{\beta}|\cdot} &= [\Sigma_{\boldsymbol{\beta}}^{-1} + \mathbf{X}^\mathrm{T} \Sigma_\mathbf{y}^{-1} \mathbf{X}]^{-1} \quad \text{and} \\ \boldsymbol{\mu}_{\boldsymbol{\beta}|\cdot} &= \Sigma_{\boldsymbol{\beta}|\cdot} \mathbf{X}^\mathrm{T} \Sigma_\mathbf{y}^{-1} \mathbf{y}\,. \end{aligned} \tag{4}$$

All the remaining parameters must be updated using Metropolis steps. Depending upon the application, this may be implemented using block-updates. On convergence, the MCMC (Markov Chain Monte Carlo) output generates L samples, say $\{\Omega^{(l)}\}_{l=1}^L$, from the posterior distribution in (3).

In updating Ω as outlined above, the spatial coefficients $\mathbf{w}$ are not sampled directly. This shrinks the parameter space resulting in a more efficient MCMC algorithm. A primary advantage of first stage Gaussian models (as in (1)) is that the posterior distribution of $\mathbf{w}$ can be recovered in a posterior predictive fashion by sampling from

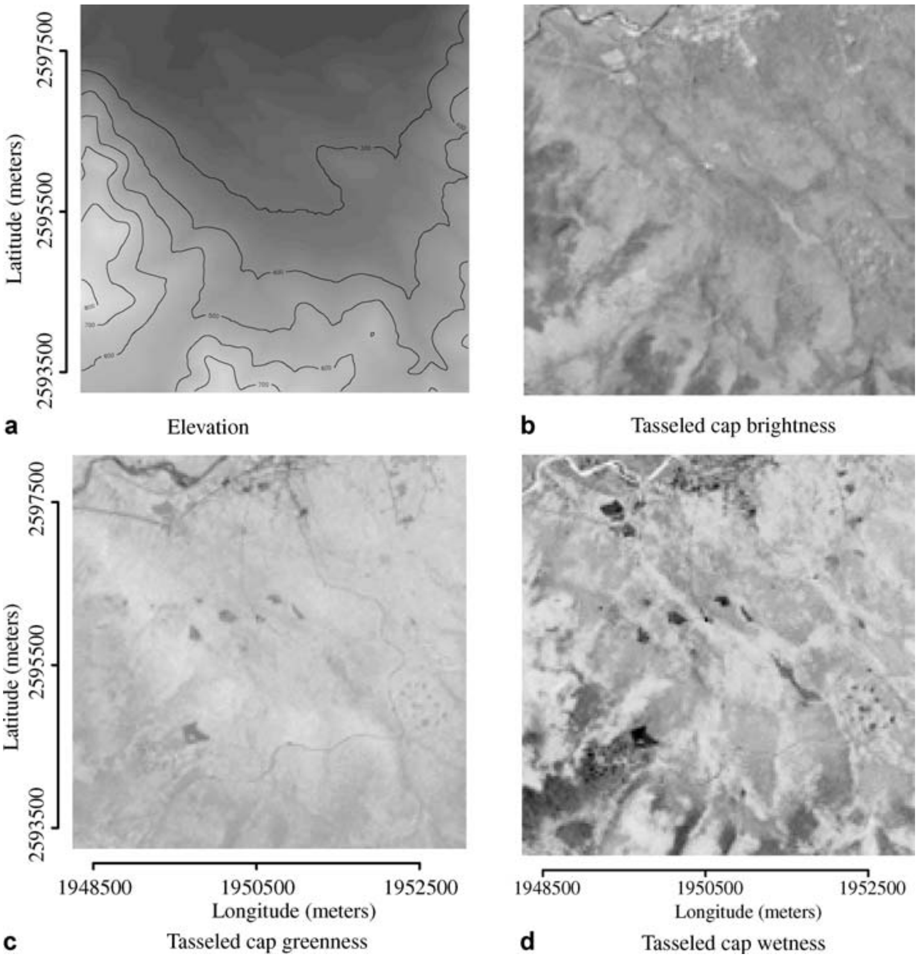
$$p(\mathbf{w} \mid \mathbf{y}) \propto \int p(\mathbf{w} \mid \Omega,\ \mathbf{y}) p(\Omega \mid \mathbf{y})\, \mathrm{d}\Omega\,. \tag{5}$$

Once the posterior samples from $p(\Omega \mid \mathbf{y})$, $\{\Omega^{(l)}\}_{l=1}^L$, have been obtained, posterior samples from $p(\mathbf{w} \mid \mathbf{y})$ are drawn by sampling $\mathbf{w}^{(l)}$ from $p(\mathbf{w} \mid \Omega^{(l)}, Data)$, one for one for each $\Omega^{(l)}$. This composition sampling is routine because $p(\mathbf{w} \mid \Omega, \mathbf{y})$ in (5) is Gaussian and the posterior estimates can subsequently be mapped with contours to produce image and contour plots of the spatial processes.

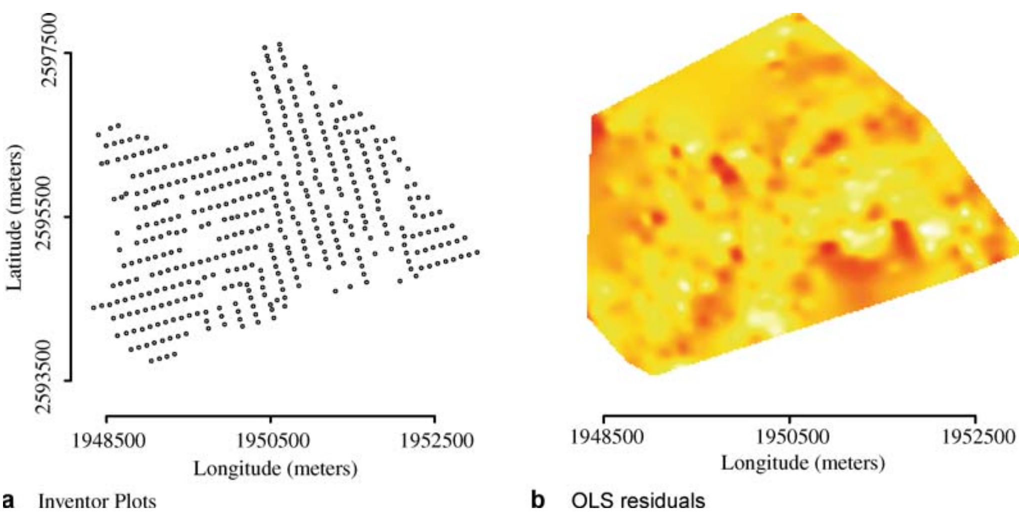
For predictions, if $\{\mathbf{s}_{0i}\}_{i=1}^{n_0}$ is a collection of n_0 locations, one can compute the posterior predictive distribution $p(\mathbf{w}^* \mid \mathbf{y})$ where $\mathbf{w}^* = [\mathbf{w}(\mathbf{s}_{0k})]_{k=1}^{n_0}$. Note that

$$p(\mathbf{w}^* \mid \mathbf{y}) \propto \int p(\mathbf{w}^* \mid \mathbf{w}, \Omega,\ \mathbf{y}) p(\mathbf{w} \mid \Omega,\ \mathbf{y}) p(\Omega \mid \mathbf{y})\, \mathrm{d}\Omega\, \mathrm{d}\mathbf{w}\,.$$

This can be computed by composition sampling by first obtaining the posterior samples $\{\Omega^{(l)}\}_{l=1}^L \sim p(\Omega \mid \mathbf{y})$, then drawing $\mathbf{w}^{(l)} \sim p(\mathbf{w} \mid \Omega^{(l)},\ \mathbf{y})$ for each l as described in (5) and finally drawing $\mathbf{w}_0^{(l)} \sim p(\mathbf{w}_0 \mid \mathbf{w}^{(l)}, \Omega^{(l)},\ \mathbf{y})$. This last distribution is derived as a conditional distribution from a multivariate normal distribution.



Bayesian Spatial Regression for Multi-source Predictive Mapping, Figure 1 Remotely sensed variables georectified to a common coordinate system (North American Datum 1983) and projection (Albers Conical Equal Area) and resampled to a common pixel resolution and alignment. The images cover the US Forest Service Bartlet Experimental Forest near Bartlett, New Hampshire USA. **a** is elevation measured in meters above sealevel derived from the 1 arc-second (approximately 30 × 30 meter) US Geological Survey national elevation dataset DEM data [17]. **b–d** are the tasseled cap components of brightness, greenness, and wetness derived from band 1–5 and 7 of a spring 2002 date of Landsat-7 ETM+ sensor imagery [20]. This Landsat imagery was acquired from the National Land-cover Database for the United States [19]



Bayesian Spatial Regression for Multi-source Predictive Mapping, Figure 2 The circle symbols in **a** represent georeference forest inventory plots on the US Forest Service Bartlett Experimental Forest near Bartlett, New Hampshire USA. **b** is an interpolated surface of residual values from a ordinary least squares regression of total tree biomass per hectare measured at forest inventory plots depicted in **a** and remotely sensed regressors, some of which are depicted in Fig. 1. Note that the spatial trends in **b** suggest that observations of total tree biomass per hectare are not conditionally independent (i. e., conditional on the regressors)

As a more specific instance, consider prediction at a single arbitrary site $\mathbf{s}_0$ evaluates $p(y(\mathbf{s}_0)|\mathbf{y})$. This latter distribution is sampled in a posterior predictive manner by drawing $y^{(l)}(\mathbf{s}_0) \sim p(y(\mathbf{s}_0)|\Omega^{(l)}, \mathbf{y})$ for each $\Omega^{(l)}, l = 1, \ldots, L$, where $\Omega^{(l)}$'s are the posterior samples. This is especially convenient for Gaussian likelihoods (such as (1)) since $P(y(\mathbf{s}_0)|\Omega^{(l)}, \mathbf{y})$ is itself Gaussian with

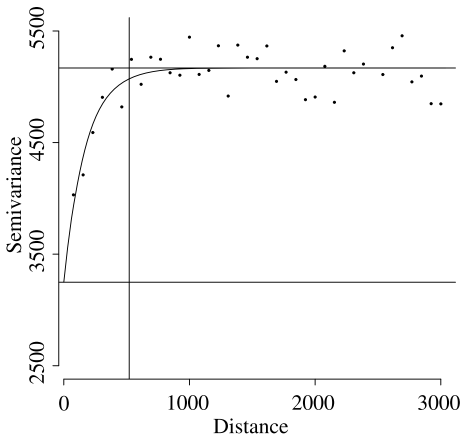
$$E[y(\mathbf{s}_0)|\Omega, \mathbf{y}] = \mathbf{x}^{\mathrm{T}}(\mathbf{s}_0)\boldsymbol{\beta} + \gamma^{\mathrm{T}}(\mathbf{s}_0) \cdot \left(R(\boldsymbol{\theta}) + \tau^2/\sigma^2 I_n\right)^{-1}(\mathbf{y} - X\boldsymbol{\beta}) \quad \text{and} \tag{6}$$

$$Var[y(\mathbf{s}_0)|\Omega, \mathbf{y}] = \sigma^2 \cdot \left(1 - \gamma^{\mathrm{T}}(\mathbf{s}_0)\left(R(\boldsymbol{\theta}) + \tau^2/\sigma^2 I_n\right)^{-1}\gamma(\mathbf{s}_0)\right) + \tau^2 , \tag{7}$$

where $\gamma(\mathbf{s}_0) = [\rho(\mathbf{s}_0, \mathbf{s}_i; \boldsymbol{\theta})]_{i=1}^N$ when $\mathbf{s}_0 \neq \mathbf{s}_i$, $i = 1, \ldots, n$ while the nugget effect τ^2 is added when $\mathbf{s}_0 = \mathbf{s}_i$ for some i. This approach is called "Bayesian Kriging."

These concepts are illustrated with data from permanent georeferenced forest inventory plots on the USDA Forest Service Bartlett Experimental Forest (BEF) in Bartlett, New Hampshire. The 1,053 hectare BEF covers a large elevation gradient from the village of Bartlett in the Saco River valley at 207 meters to about 914 meters above sea level. For this illustration, the focus is on predicting the spatial distribution of total tree biomass per hectare across the BEF. Tree biomass is measured as the weight of all above ground portions of the tree, expressed here as metric tons per hectare. Within the data set, biomass per hectare is recorded at 437 forest inventory plots across the BEF, Fig. 2. Satellite imagery and other remotely sensed variables have proved useful regressors for predicting forest biomass. A spring, summer, and fall 2002 date of 30×30 Landsat 7 ETM+ satellite imagery was acquired for the BEF. Following [20], the image was transformed to tasseled cap components of brightness (1), greenness (2), and wetness (3) using data reduction techniques. Three of the nine resulting spectral variables labeled TC1, TC2, and TC3 are depicted in Fig. 1b–d. In addition to these nine spectral variables, digital elevation model data was used to produce a 30×30 elevation layer for the BEF Fig. 1a. The centroids of the 437 georeferenced inventory plots were intersected with the elevation and spectral variables to form the 437×11 covariate matrix (i. e., an intercept, elevation, and nine spectral components).

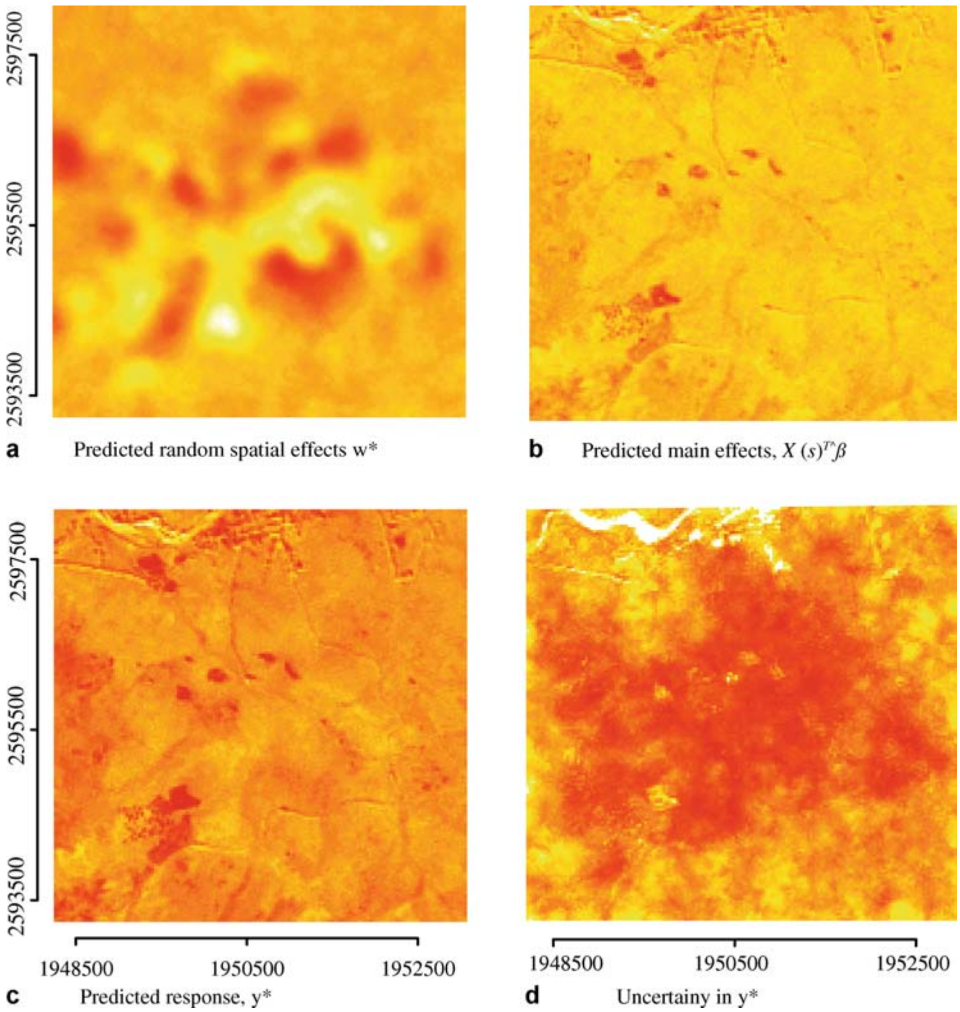
Choice of priors is often difficult and therefore in practice it is helpful to initially explore the data with empirical semivariograms and density plots. Examination of the semivariogram in Fig. 3, suggests an appropriate prior would center the spatial variance parameter σ^2 at ~2,000 (i. e., the distance between the lower and upper horizontal lines), the nugget or measurement error parameter τ^2 at



Bayesian Spatial Regression for Multi-source Predictive Mapping, Figure 3 Empirical semivariograms and exponential function REML parameter estimates of residual values from a ordinary least squares regression of total tree biomass per hectare measured at forest inventory plots depicted in Fig. 2 and remotely sensed regressors Fig. 1. The estimates of the nugget (*bottom horizontal line*), sill (*upper horizontal line*), and range (*vertical line*) are approximately 3250, 5200, and 520 respectively

~3,000, and a support of 0–1,500 for the spatial range ϕ. There are several popular choices of priors for the variance parameters including inverse-Gamma, half-Cauchy, and half-normal (see, e. g., [15]). Commonly, a uniform prior is placed on the spatial range parameter. Once priors are chosen, the Bayesian specification is complete and the model can be fit and predictions made as described above. Predictive maps are depicted in Fig. 4. Here, the mean of the posterior predictive distribution of the random spatial effects, $E[\mathbf{w}^* \mid \mathbf{y}]$, is given in a. The mean predicted main effect over the domain, $\mathbf{X}\widehat{\boldsymbol{\beta}}$, where $\widehat{\boldsymbol{\beta}} = E[\boldsymbol{\beta} \mid \mathbf{y}]$ is depicted in (b) and c gives the mean predictive surfaces, $E[\mathbf{y}^* \mid \mathbf{y}]$, where $\mathbf{y}^* = \mathbf{X}\widehat{\boldsymbol{\beta}} + \mathbf{w}^*$. The uncertainty in surface c is summarized in d as the range between the upper and lower 95% credible interval for each pixel's predictive distribution. Note, the trend in pixel-level uncertainty in d is consistent with the inventory plot locations in Fig. 2, specifically uncertainty in prediction increases with increasing distance from observed sites.

In practical data analysis, it is more common to explore numerous alternative models representing different hypothesis and perhaps incorporating varying degrees of spatial richness. This brings up the issue of comparing these models and perhaps ranking them in terms of better performance. Better performance is usually judged employing posterior predictive model checks with predictive samples and perhaps by computing a global measure of fit. One popular approach proceeds by computing the posterior predictive distribution of a *replicated data set*, $p(\mathbf{y}_{\mathrm{rep}}|\mathbf{y}) = \int p(\mathbf{y}_{\mathrm{rep}}|\Omega)p(\Omega|\mathbf{y})\,\mathrm{d}\Omega$, where $p(\mathbf{y}_{\mathrm{rep}}|\Omega)$ has the same distribution as the data likelihood. Replicated



Bayesian Spatial Regression for Multi-source Predictive Mapping, Figure 4 Results of pixel-level prediction from a spatial regression model of the response variable total tree biomass per hectare, depicted in Fig. 2, and remotely sensed regressors, some of which are depicted in Fig. 1

data sets from the above distribution are easily obtained by drawing, for each posterior realization $\Omega^{(l)}$, a replicated data set $\mathbf{y}_{\text{rep}}^{(l)}$ from $p(\mathbf{y}_{\text{rep}}|\Omega^{(l)})$. Preferred models will perform well under a decision-theoretic *balanced loss function* that penalizes both departure from corresponding observed value (lack of fit), as well as for what the replicate is expected to be (variation in replicates). Motivated by a squared error loss function, the measures for these two criteria are evaluated as $G = (\mathbf{y} - \boldsymbol{\mu}_{\text{rep}})^{\mathrm{T}}(\mathbf{y} - \boldsymbol{\mu}_{\text{rep}})$ and $P = tr(Var(\mathbf{y}_{\text{rep}}|\mathbf{y}))$, where $\boldsymbol{\mu}_{\text{rep}} = E[\mathbf{y}_{\text{rep}}|\mathbf{y}]$ is the posterior predictive mean for the replicated data points, and P is the trace of the posterior predictive dispersion matrix for the replicated data; both these are easily computed from the samples $\mathbf{y}_{\text{rep}}^{(j)}$. [14] suggests using the score $D = G + P$ as a model selection criterion, with lower values of D indicating better models.

Another measure of model choice that has gained much popularity in recent times, especially due to computational convenience, is the Deviance Information Criteria (DIC) [34]. This criteria is the sum of the Bayesian deviance (a measure of model fit) and the (effective) number of parameters (a penalty for model complexity). The deviance, up to an additive quantity not depending upon Ω, is $D(\Omega) = -2\log L(\mathbf{y}|\ \Omega)$, where $L(\mathbf{y}|\ \Omega)$ is the first stage Gaussian likelihood as in (1). The Bayesian deviance is the posterior mean, $\overline{D(\Omega)} = E_{\Omega|\mathbf{y}}[D(\Omega)]$, while the effective number of parameters is given by $p_D = \overline{D(\Omega)} - D(\bar{\Omega})$. The DIC is then given by $\overline{D(\Omega)} + p_D$ and is easily computed from the posterior samples. It rewards better fitting models through the first term and penalizes more complex models through the second term, with lower values indicating favorable models for the data.

Key Applications

Risk Assessment

Spatial and/or temporal risk mapping and automatic zonation of geohazards have been modeled using traditional geostatistical techniques that incorporate both raster and vector data. These investigations attempt to predict the spatial and temporal distribution of risk to humans or components of an ecosystem. For example, [35] explore the utility of geostatistics for human risk assessments of hazardous waste sites. Another example is from [23] who investigate the uncertainty of ecological risk estimates concerning important wildlife species. As noted, the majority of the multi-source risk prediction literature is based on non-Bayesian kriging models; however, as investigators begin to recognize the need to estimate the uncertainty of prediction, they will likely embrace the basic Bayesian methods reviewed here, and extend them to fit their specific domain. For example, [22] have proposed one such Bayesian extension to spatially explicit hazard regression.

Agricultural and Ecological Assessment

Spatial processes, such as predicting agricultural crop yield and environmental conditions (e. g., deforestation, soil or water pollution, or forest species change in response to changing climates) are often modeled using multi-source spatial regression (see., e. g., [1,5,6]). Only recently have Bayesian models been used for predicting agricultural and forest variables of interest within a multi-source setting. For example, in an effort to quantify forest carbon reserves, [3] used single and multiple resolution Bayesian spatial regression to predict the distribution of forest biomass. An application of such models to capture spatial variation in growth patterns of weeds is discussed in [4].

Atmospheric and Weather Modeling

Arrays of weather monitoring stations provide a rich source of spatial and temporal data on atmospheric conditions and precipitation. These data are often coupled with a host of topographic and satellite derived variables through a spatial regression model to predict short- and long-term weather conditions. Recently, several investigators used these data to illustrate the virtues of a Bayesian approach to spatial prediction (see, [12,27,28]).

Future Directions

Over the last decade hierarchical models implemented through MCMC methods have become especially popular for spatial modeling, given their flexibility to estimate models that would be infeasible otherwise. However, fitting hierarchical spatial models often involves expensive matrix decompositions whose complexity increases exponentially with the number of spatial locations. In a fully Bayesian paradigm, where one seeks formal inference on the correlation parameters, these matrix computations occur in each iteration of the MCMC rendering them completely infeasible for large spatial data sets. This situation is further exacerbated in multivariate settings with several spatially dependent response variables, where the matrix dimensions increase by a factor of the number of spatially dependent variables being modeled. This is often referred to as the "big *N* problem" in spatial statistics, and has been receiving considerable recent attention. While attempting to reduce the dimension of the problem, existing methods incur certain drawbacks such as requiring the sites to lie on a regular grid, which entail realigning irregular spatial data using an algorithmic approach that might lead to unquantifiable errors in precision.

Another area receiving much attention in recent times is that of multivariate spatial models with multiple response variables as well as dynamic spatiotemporal models. With several spatially dependent variables the association between variables accrue further computational costs for hierarchical models. More details about multivariate spatial modeling can be found in [2,9,10,36]. A special class of multivariate models known as coregionalization models is discussed and implemented in [13]. Unless simplifications such as *separable* or *intrinsic* correlation structures (see [36]) are made, multivariate process modeling proves too expensive for reasonably large spatial datasets. This is another area of active investigation.

Cross References

- ► Data Analysis, Spatial
- ► Hierarchical Spatial Models
- ► Hurricane Wind Fields, Multivariate Modeling
- ► Kriging
- ► Public Health and Spatial Modeling
- ► Spatial Regression Models
- ► Spatial Uncertainty in Medical Geography: A Geostatistical Perspective
- ► Statistical Descriptions of Spatial Patterns
- ► Uncertainty, Modeling with Spatial and Temporal

Recommended Reading

1. Atkinson, P.M., Webster, R., Curran, P.J.: Cokriging with airborne MSS imagery. Remote Sens. Environ. **50**, 335–345 (1994)
2. Banerjee, S., Carlin, B.P., Gelfand, A.E.: Hierarchical Modeling and Analysis for Spatial Data. Chapman and Hall/CRC Press Boca Raton, FL: (2004)
3. Banerjee, S., Finley, A.O.: Bayesian multi-resolution modelling for spatially replicated datasets with application to forest biomass data. J. Statistical Plann. Inference (2007). doi: 10.1016/j.jspi.2006.05.024
4. Banerjee, S., Johnson, G.A.: Coregionalized single and multi-resolution spatially varying growth-curve modelling. Biometrics, **62**, 864–876 (2006)
5. Berterretche, M., Hudak, A.T., Cohen, W.B., Maiersperger, T.K., Gower, S.T., Dungan, J.: Comparison of regression and geosta-

tistical methods for mapping Leaf Area Index (LAI) with Landsat ETM+ data over a boreal forest. Remote Sens. Environ. **96**, 49–61 (2005)
6. Bhatti, A.U., Mulla, D.J., Frazier, B.E.: Estimation of soil properties and wheat yields on complex eroded hills using geostatistics and thematic mapper images. Remote Sensing Environ. **37**(3), 181–191 (1991)
7. Carlin, B.P., Louis, T.A.: Bayes and Empirical Bayes Methods for Data Analysis. 2nd edn. Chapman and Hall/CRC Press Boca Raton, FL: (2000)
8. Campbell, J.B.: Introduction to Remote Sensing, 4th edn. The Guilford Press, New York (2006) p 626
9. Chilées, J.P., Delfiner, P.: Geostatistics: Modelling Spatial Uncertainty. John Wiley and Sons, New York (1999)
10. Cressie, N.A.C.: Statistics for Spatial Data, 2nd edn. Wiley, New York (1993)
11. Cromley, E.K., McLafferty, S.L.: GIS and Public Health. Guilford Publications, Inc, New York (2002)
12. Diggle, P.J., Ribeiro Jr, P.J.: Bayesian Inference in Gaussian Model-based Geostatistics. Geogr. Environ. Modelling **6**, 29–146 (2002)
13. Finley, A.O., Banerjee, S., Carlin, B.P.: spBayes: An R package for univariate and multivariate hierarchical point-referenced spatial models. J. Statistical Softw. **19**, 4 (2007)
14. Gelfand, A.E., Ghosh, S.K.: Model Choice: a minimum posterior predictive loss approach. Biometrika. *85*, 1–11 (1998)
15. Gelman, A.: Prior distributions for variance parameters in hierarchical models. Bayesian Anal. **3**, 515–533 (2006)
16. Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B.: Bayesian Data Analysis, 2nd edn. Chapman and Hall/CRC Press Boca Raton, FL (2004)
17. Gesch, D., Oimoen, M., Greenlee, S., Nelson, C., Steuck, M., Tyler, D.: The national elevation dataset. Photogrammetric Eng. Remote Sensing **68**(1), 5–12 (2002)
18. Handcock, M.S., Stein, M.L.: A Bayesian analysis of kriging. Technometrics. **35**, 403–410 (1993)
19. Homer C., Huang C., Yang L., Wylie, B., Coan, M.: Development of a 2001 National Land-cover Database for the United States. Photogramm. Eng. Remote Sens. **70**, 829–840 (2004)
20. Huang C., Wylie B., Homer C., Yang L., Zylstra, G.: Derivation of a Tasseled Cap Transformation Based on Landsat 7 At-Satellite Reflectance. Int. J. Remote Sens. **8**, 1741–1748 (2002)
21. Jones, C.B.: Geographical Information Systems and Computer Cartography. Addison Wesley Longman, Harlow, Essex, UK (1997)
22. Kneib, T., Fahrmeir, L.: A Mixed Model Approach for Geoadditive hazard Regression. Scandinavian J. Stat. **34**, 207–228 (2007)
23. Kooistra, L., Huijbregts, M.A.J., Ragas, A.M.J., Wehrens, R., Leuven, R.S.E.W.: Spatial Variability and Uncertainty in Ecological Risk Assessment: A Case Study on the Potential Risk of Cadmium for the Little Owl in a Dutch River Flood Plain. Environ. Sci. Technol. **39**, 2177–2187 (2005)
24. Möller, J.: Spatial Statistics and Computational Method. Springer, New York (2003)
25. Mather, P.M.: Computer Processing of Remotely-Sensed Images, 3rd edn. John Wiley and Sons, Hoboken, NY (2004) p 442
26. National Academy of Sciences: Remote Sensing with Special Reference to Agriculture and Forestry. Washington, DC: National Academy of Sciences, (1970) p 424
27. Paciorek, C.J., Schervish, M.J.: Spatial modelling using a new class of nonstationary covariance functions. Environ. **17**, 483–506 (2006)
28. Riccio, A., Barone, G., Chianese, E., Giunta, G.: A hierarchical Bayesian approach to the spatio-temporal modeling of air quality data. Atmosph. Environ. **40**, 554–566 (2006)
29. Richards, J.A., Xiuping, J.: Remote Sensing Digital Image Analysis, 4th edn. Springer-Verlag, Heidelberg (2005) p 439
30. Robert, C.: The Bayesian Choice, 2nd edn. Springer, New York (2001)
31. Santner, T.J., Williams, B.J., Notz, W.I.: The Design and Analysis of Computer Experiments. Springer, New York (2003)
32. Schabenberger, O., Gotway, C.A.: Statistical Methods for Spatial Data Analysis (Texts in Statistical Science Series). Chapman and Hall/CRC, Boca Raton (2004)
33. Scheiner, S.M., Gurevitch, J.: Design and Analysis of Ecological Experiments, 2nd edn. Oxford University Press, Oxford, UK (2001)
34. Spiegelhalter, D.J., Best, N.G., Carlin, B.P., van der Linde, A.: Bayesian measures of model complexity and fit (with discussion and rejoinder). J. R. Statistical Soc. Series B, **64**, 583–639 (2002)
35. Thayer, W.C., Griffith, D.A., Goodrum, P.E., Diamond, G.L., Hassett, J.M.: Application of Geostatistics to Risk Assessment. Risk Analysis: An Int. J. **23**(5), 945–960 (2003)
36. Wackernagel, H.: Multivariate Geostatistics: An introduction with Applications, 3nd edn. Springer-Verlag, New York (2006)
37. Wang, Y., Zheng, T.: Comparison of Light Detection and Ranging and National Elevation Dataset Digital Elevation Model on Floodplains of North Carolina. Nat. Hazards Rev. **6**(1), 34–40 (2005)
38. Webster, R., Oliver, M.A.: Geostatistics for Environmental Scientists. John Wiley and Sons, New York (2001)

Best Linear Unbiased Prediction

► Spatial Econometric Models, Prediction

Bi-Temporal

► Spatio-temporal Query Languages

Bioinformatics, Spatial Aspects

SUDHANSHU SEKHAR PANDA
Department of Science, Engineering, and Technology,
Gainesville State College, Gainsville, GA, USA

Synonyms

Biological data mining; Epidemiological mapping; Genome mapping

Definition

The web definition suggests that bioinformatics is the science that deals with the collection, organization and analysis of large amounts of biological data using advanced information technology such as networking of computers, software, and databases. This is the field of science in which biology, computer science, and information technology merge into a single discipline. Bioinformatics deals with information about human and other animal genes and related biological structures and processes. Genome structures such as DNA, RNA, proteins in cells, etc., are spatial in nature (Fig. 1a, b and c). They can be represented as 2- or 3-dimensional.

Because of the spatial nature of genome data, geographic information systems (GIS), being an information science, has a larger role to play in the area of bioinformatics. GIS and bioinformatics have much in common. Digital maps, large databases, visualization mapping, pattern recognition and analysis, etc., are common tasks in GIS and bioinformatics [1]. While GIS research is based on analyzing satellite or aerial photos, gene research deals with high-resolution images (Fig. 2) generated by microscopes or laboratory sensing technologies. GIS techniques and tools are used by researchers for pattern recognition, e. g., geographic distribution of cancer and other diseases in human, animal, and plant populations [1,2].

Historical Background

According to Kotch [2], spatial epidemiology mapping and analysis has been driven by software developments in Geographic Information Systems (GIS) since the early 1980's. GIS mapping and analysis of spatial disease patterns and geographic variations of health risks is helping understand the spatial epidemiology since past centuries [3]. Researchers in bioinformatics also deal with similar pattern recognition and analysis regarding very small patterns, such as those in DNA structure that might predispose an organism to developing cancer [1]. As both bioinformatics and GIS are based on common mapping and analytical approaches, there is a good possibility of gaining an important mechanistic link between individual-level processes tracked by genomics and proteomics and population-level outcomes tracked by GIS and epidemiology [1]. Thus, the scope of bioinformatics in health research can be enhanced by collaborating with GIS.

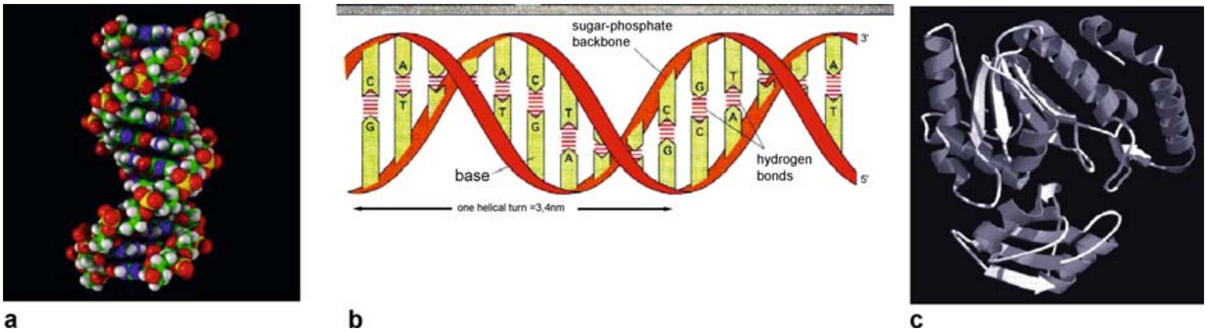
B

Scientific Fundamentals

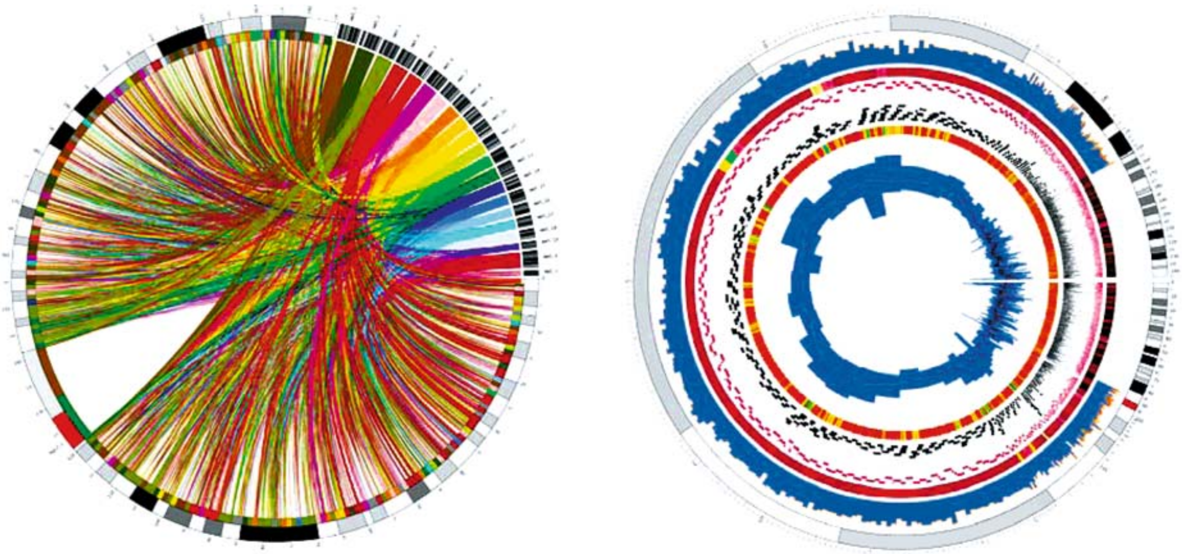
As discussed earlier, data in bioinformatics are of spatial nature and could be well understood if represented, analyzed, and comprehended just like other geo-spatial data. GIS can interactively be used in bioinformatics projects for better dynamism, versatility, and efficiency. Figure 3 shows mapping of genome data using ArcGIS software. This helps in managing the genome data interactively with the application of superior GIS functionality. Below is a description of a GIS application in bioinformatics for different aspects of management and analysis.

Use of GIS for Interactive Mapping of Genome Data

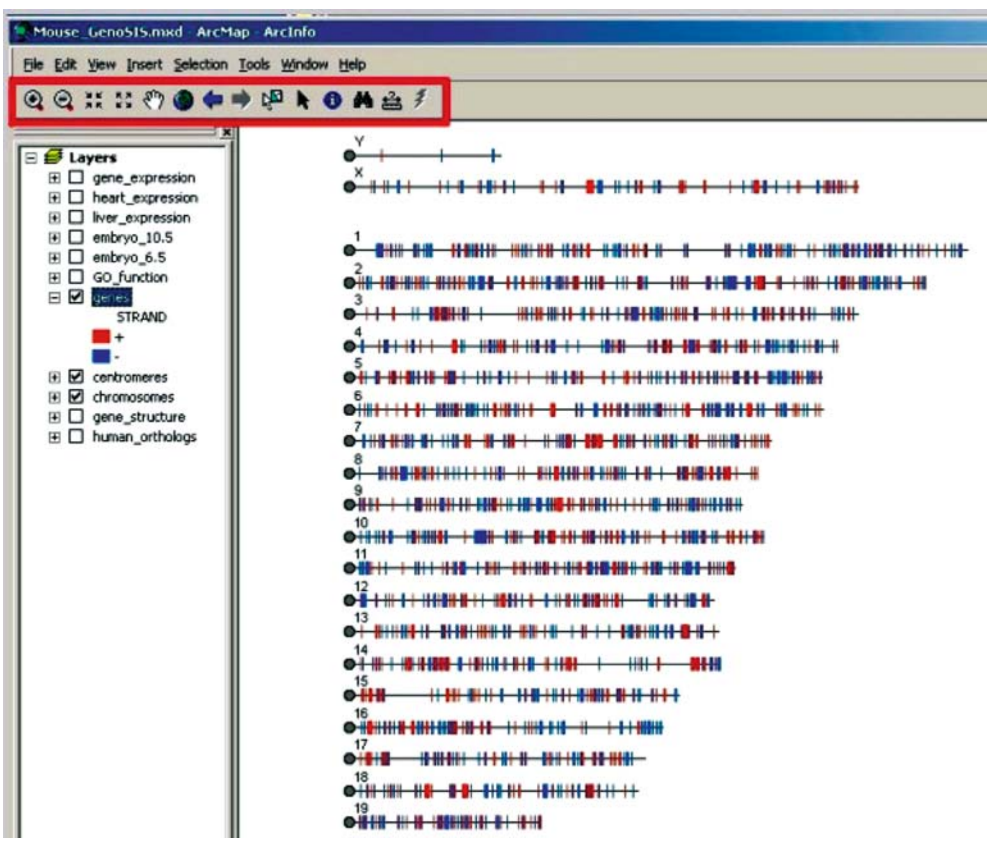
In bioinformatics application, genome browsers are developed for easy access of the data. They use only simple keyword searches and limit the display of detailed annotations to one chromosomal region of the genome at a time [4]. Spatial data browsing and management could be done with efficiency using ArcGIS software [5]. Dolan et al. [4] have employed concepts, methodologies, and the tools that were developed for the display of geographic data to develop a Genome Spatial Information System (GenoSIS) for spatial display of genomes (Fig. 4). The GenoSIS helps users to dynamically interact with genome annotations and related attribute data using query tools of ArcGIS, such as query by attributes, query by location, query by graphics, and developed definition queries. The project also helps in producing dynamically generated genome maps for users. Thus, the application of GIS in bioinformatics projects helps genome browsing become more versatile and dynamic.



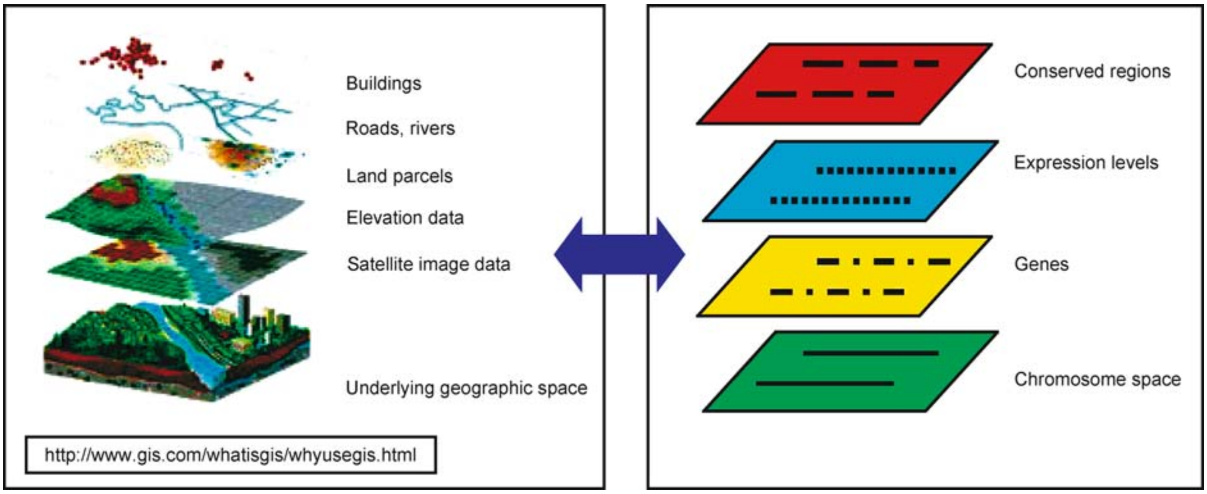
Bioinformatics, Spatial Aspects, Figure 1 Images of DNA and proteins in cell as 2-D and 3-D form



Bioinformatics, Spatial Aspects, Figure 2 High-resolution (4000 × 4000 pixels) images of a genome maps showing the spatial nature of the data



Bioinformatics, Spatial Aspects, Figure 3 Display of a mouse genome in ArcGIS (adapted from Dolan et al. 2006 [4])



Bioinformatics, Spatial Aspects, Figure 4 Comparative use of GIS paradigm of the map layers to the integration and visualization of genome data (adapted from Dolan et al. 2006 [4])

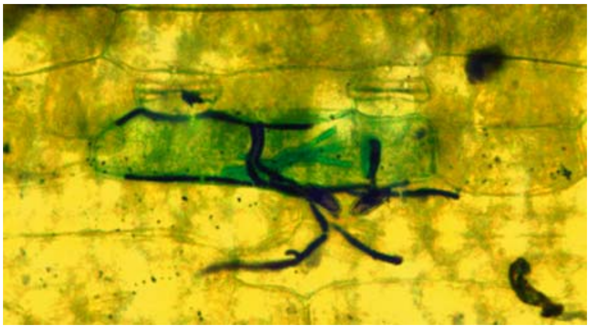
GIS Application as a Database Tool for Bioinformatics

GIS can be applied for efficient biological database management. While developing a database for the dynamic representation of marine microbial biodiversity, the GIS option provided Pushker et al. [6] with an interface for selecting a particular sampling location on the world map and getting all the genome sequences from that location and their details. Geodatabase management ability helped them obtained the following information: i) taxonomy report: taxonomic details at different levels (domain, phylum, class, order, family and genus); ii) depth report: a plot showing the number of sequences vs. depth; iii) biodiversity report: a list of organisms found; iv) get all entries and v) advanced search for a selected region on the map. Using GIS tools, they retrieved sequences corresponding to a particular taxonomy, depth or biodiversity[6]. Meaning, the bioinformatics dealing with a spatial scale can be well managed by GIS database development and management.

While developing a "Global Register of Migratory Species (GROMS)", Riede [7] developed a Geodatabase of the bird features including their genomes. It was efficient in accessing the information about the migratory birds including their spatial location through the Geodatabase.

Genome Mapping and Pattern Recognition and Analysis

While studying the genome structure, it is essential to understand the spatial extent of its structure. As genome mapping is done through imaging, image processing tools are used to analyze the pattern. GIS technology could be used for pattern recognition and analysis. Figure 5 is an example of an image of a microscopic top view of a barley leaf with a transformed β-Glucuronidase-expressing cell (blue-green). From the image analysis, it is observed that two fungal spores of *Blumeria graminis f.sp. hordei* (fungus dark blue) are interacting with the transformed cell and the spore at the left hand side successfully penetrated into the transformed cell and started to grow out on the leaf surface illustrated by the elongating secondary hyphae. This study shows the spatial aspect of bioinformatics [8].



Bioinformatics, Spatial Aspects, Figure 5 Image analysis of a Barley leaf for cell transformation analysis (adapted from Schweizer 2007 [8])

GIS Software in Bioinformatics as Spatial Analysis Tool

Software has been developed as tools of bioinformatics to analyze nucleotide or amino acid sequence data and extract biological information. 'Gene prediction software [9]' and 'Sequence alignment software[10]' are examples of some of the software developed for bioinformatics.

Gene prediction software is used to identify a gene within a long gene sequence. As described by Dolan et al. [4], if the genome database can be presented through ArcGIS, they can be visualized, analyzed, and queried better than the present available techniques. Thus, GIS function development techniques can be replicated to make the available software more efficient. Programs like GenoSIS [4] is a step in that direction. Sequence alignment software is a compilation of bioinformatics software tools and web portals which are used in sequence alignment, multiple sequence alignment, and structural alignment [10]. They are also used for database searching. Thus, GIS can be used for bringing dynamism to the database search. Matching of DNA structures could be efficiently done with the GIS application.

Molecular modeling and 3-D visualization is another aspect of genome research. To understand the function of proteins in cells, it is essential to determine a protein's structure [11]. The process of determining a protein's exact structure is labor intensive and time consuming. Traditionally, X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy techniques are used to solve protein structure determination [11]. The maps developed by these instruments are preserved in the form of a Protein Data Bank (PDB). The PDB is the first bioinformatics resource to store three-dimensional protein structures. Currently, it is possible to visualize the utility of GIS regarding molecular modeling and the 3-D visualization process. ArcScence™ is the 3-D visualization and modeling software for spatial data. It could very well be the best tool for PDB data visualization, modeling, and analysis.

Key Applications

Bioinformatics is playing important roles in many areas such as agriculture, medicine, biotechnology, environmental science, animal husbandry, etc., as a genome is not only a principal component of the human body but also of plants. GIS or geotechnology has been successfully used in these areas for a long time. Therefore, using bioinformatics in these areas could be associated with spatial technology of GIS. A study by Nielsen and Panda [12] was conducted on predictive modeling and mapping of fasting blood glucose level in Hispanics in southeastern Idaho. The levels were mapped according to the racial genes and their spatial aspect of representation. This study shows how bioinformatics can be used in several other areas including epidemiology.

Future Directions

According to Virginia Bioinformatics Institute (VBI) Director Bruno Sobral, "The notion of a map goes all the way from the level of a genome to a map of the United States," he said, "bioinformatics has focused on modeling from the level of the molecules up to the whole organism, while GIS has created tools to model from the level of the ecosystem down." This indicates that there is great potential for bioinformatics and geospatial technology to be combined in a mutually enhancing fashion for important applications.

Cross References

► Biomedical Data Mining, Spatial
► Public Health and Spatial Modeling

Recommended Reading

1. Anonymous: VT Conference Puts New Research Area on the Map; GIS Expert Michael Goodchild Echoes Its Value. https://www.vbi.vt.edu/article/articleview/48/1/15/ (2007). Accessed 05 March 2007
2. Kotch, T.: Cartographies of Disease: Maps, Mapping and Disease. ESRI Press, Redlands, CA (2005)
3. Jacquez, G.M.: Spatial Analysis in Epidemiology: Nascent Science or a Failure of GIS? J Geo Sys **2**,91–97 (2000)
4. Dolan, M.E., Holden, C.C., Beard, M.K., Bult, C.J.: Genomes as geography: using GIS technology to build interactive genome feature maps. BMC Bioinf 7:416 (2006)
5. ESRI. ArcIMS. http://www.esri.com/software/arcgis/arcims/index.html. Accessed 05 March 2007
6. Pushker, R., D'Auria, G., Alba-Casado, J.C., Rodríguez-Valera, F.: Micro-Mar: a database for dynamic representation of marine microbial biodiversity. BMC Bioinf 6:222 (2005)
7. Riede, K.: Conservation and modern information technologies: The Global Register of Migratory Species (GROMS). J Int Wildl Law Policy **3**(2), 152–165 (2000)
8. Schweizer, P.: IPK Gatersleben. www.ipk-gatersleben.de/en/02/04/04/ (2007). Accessed 05 March 2007
9. Pavy, N., Rombauts, S., Dehais, P, Mathe, C., Ramana, D.V.V., Leroy, P., Rouze, P.: Evaluation of gene prediction software using a genomic data set: application to Arabidopsis thaliana sequence. Bioinf **15**(11), 887–899 (1999)
10. Anonymous: Sequence alignment software. http://en.wikipedia.org/wiki/Sequence_alignment_software (2007). Accessed
11. BSCS. Bioinformatics and the human genome project. A Curriculum Supplement for High School Biology. Developed by BSCS under a contract from the Department of Energy (2003)
12. Nielson, J., Panda, S.S.: Predictive modeling and mapping of fasting blood glucose: A preliminary study of diabetes among Hispanics in SE Idaho. Presented in the Intermountain GIS User's Conference, 4–8 April 2006, Helena, Montana, USA (2006)

Biological Data Mining

► Bioinformatics, Spatial Aspects

B

Biomedical Data Mining, Spatial

KEITH MARSOLO[1], MICHAEL TWA[2], MARK A. BULLIMORE[2], SRINIVASAN PARTHASARATHY[1]
[1] Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, USA
[2] College of Optometry, The Ohio State University, Columbus, OH, USA

Synonyms

Polynomials, orthogonal; Polynomials; Zernike; Zernike polynomials; Wavelets

Definition

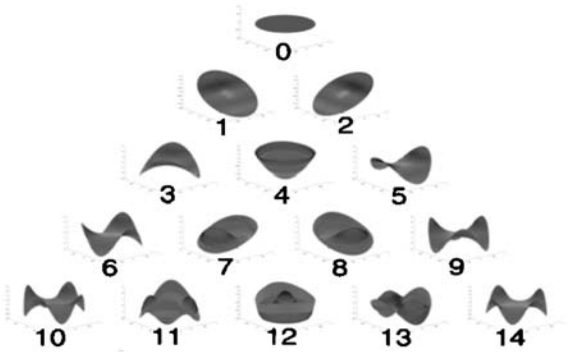
The use of biomedical data in object classification presents several challenges that are well-suited to knowledge discovery and spatial modeling methods. In general, this problem consists of extracting useful patterns of information from large quantities of data with attributes that often have complex interactions. Biomedical data are inherently multidimensional and therefore difficult to summarize in simple terms without losing potentially useful information. A natural conflict exists between the need to simplify the data to make it more interpretable and the associated risk of sacrificing information relevant to decision support.

Transforming spatial features in biomedical data quantifies, and thereby exposes, underlying patterns for use as attributes in data mining exercises [14]. To be useful, a data transformation must faithfully represent the original spatial features. Orthogonal polynomials such as the Zernike polynomials have been successfully used to transform very different types of spatial data [3,17]. The use of wavelets [8], coupled with space-filling curves [12] remains another possibility of representing spatial data.

The following entry presents general findings from work in the use of spatial modeling methods to represent and classify the shape of the human cornea.

Historical Background

Zernike polynomials were originally derived to describe optical aberrations and their geometric modes have a direct relation to the optical function of the eye [1,17]. Several of the circular polynomial modes of Zernike polynomials, presented in Fig. 1 show strong correlation with natural anatomical features of the cornea (i. e. normal corneal asphericity and astigmatic tori-city). Zernike polynomials have been used within the medical community for a number of different applications, most recently for modeling the shape of the cornea [13]. Studies have shown that these models can effectively characterize aberrations that may exist on the corneal surface [4,5].



Biomedical Data Mining, Spatial, Figure 1 Graphical representation of the modes that constitute a 4th order Zernike polynomial. Each mode is labeled using the single indexing convention created by the Optical Society of America [15]

The use of wavelets is natural in applications that require a high degree of compression without a corresponding loss of detail, or where the detection of subtle distortions and discontinuities is crucial [8]. Wavelets have been used in a number of applications, ranging from signal processing, to image compression, to numerical analysis [2]. They play a large role in the processing of biomedical instrument data obtained through techniques such as ultrasound, magnetic resonance imaging (MRI) and digital mammography [7].

Scientific Fundamentals

Zernike Polynomials

Zernike polynomials are a series of circular polynomials defined within the unit circle. They are orthogonal by the following condition:

$$\int_0^{2\pi}\int_0^1 Z_n^m(\rho,\theta) ZR_{n'}^{m'}(\rho,\theta)\rho\,\mathrm{d}\rho\,\mathrm{d}\theta = \frac{\pi}{2(n+1)}\delta nn'\delta mm' \tag{1}$$

where δ is the Kronecker delta. Their computation results in a series of linearly independent circular geometric modes that are orthonormal with respect to the inner product given above. Zernike polynomials are composed of three elements: a normalization coefficient, a radial polynomial component and a sinusoidal angular component [15]. The general form for Zernike polynomials is given by:

$$Z_n^m(\rho,\theta) = \begin{cases} \sqrt{2(n+1)}ZR_n^m(\rho)\cos(m\theta) & \text{for } m > 0 \\ \sqrt{2(n+1)}ZR_n^m(\rho)\sin(|m|\theta) & \text{for } m < 0 \\ \sqrt{(n+1)}ZR_n^m(\rho) & \text{for } m = 0 \end{cases} \tag{2}$$

where n is the radial polynomial order and m represents azimuthal frequency. The normalization coefficient is given by the square root term preceding the radial and azimuthal components. The radial component of the Zernike polynomial, the second portion of the general formula, is defined as:

$$ZR_n^m(\rho) = \sum_{s=0}^{(n-|m|)/2} \frac{(-1)^s (n-s)!}{s!(\frac{n+|m|}{2}-s)!(\frac{n-|m|}{2}-s)!} \rho^{n-2s}. \quad (3)$$

Note that the value of n is a positive integer or zero. For a given n, m can only take the values $-n$, $-n+2$, $-n+4$, $\ldots$, n. In other words, $m - |n| = even$ and $|n| < m$. Thus, only certain combinations of n and m will yield valid Zernike polynomials. Any combination that is not valid simply results in a radial polynomial component of zero.

Polynomials that result from fitting raw data with these functions are a collection of approximately orthogonal circular geometric modes. The coefficients of each mode are proportional to its contribution to the overall topography of the original image data. As a result, one can effectively reduce the dimensionality of the data to a subset of polynomial coefficients that represent spatial features from the original data as the magnitude of discrete orthogonal geometric modes.

Wavelets

Given a decomposition level S and a one-dimensional signal of length N, where N is divisible by 2^S, the *discrete wavelet transform* consists of S stages. At each stage in the decomposition, two sets of coefficients are produced, the approximation and the detail. The approximation coefficients are generated by convolving the input signal with a *low-pass* filter and down-sampling the results by a factor of two. The detail coefficients are similarly generated, convolving the input with a *high-pass* filter and down-sampling by a factor of two. If the final stage has not been reached, the approximation coefficients are treated as the new input signal and the process is repeated. In many cases, once the wavelet transformation is complete, the original signal will be represented using combinations of approximation and detail coefficients. In the applications presented here, however, *only the final level of approximation coefficients are used.* The rest are simply ignored.

To operate on a two-dimensional signal of size $N \times N$ (with N divisible by 2^S), the decomposition proceeds as follows: First, the rows are convolved with a low-pass filter and down-sampled by a factor of two, resulting in matrix L ($N/2 \times N$). The process is repeated on the original signal using a high-pass filter, which leads to matrix H ($N/2 \times N$). The columns of L are convolved two separate times, once with a low-pass filter and again with a high-pass filter. After passing through the filters, the signals are down-sampled by a factor of two. This results in a matrix of approximation (LL) and horizontal detail coefficients (LH), respectively (both of size $N/2$ x $N/2$). These steps are executed once more, this time on the columns of H, resulting in a matrix of diagonal (HH) and vertical (HL) detail coefficients (again of size $N/2$ x $N/2$). The whole procedure can then be repeated on the approximation coefficients contained in matrix LL. As in the one-dimensional case, only the final level of approximation coefficients are considered.

Space-Filling Curves

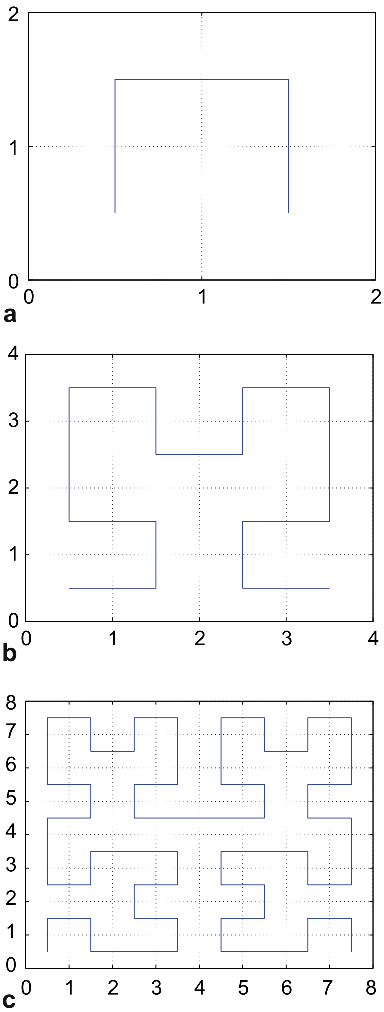
The two-dimensional wavelet decomposition will, to some degree, capture the spatial locality of the original data. There may be instances, however, where the data does not lend itself to the two-dimensional decomposition and the one-dimensional transformation is preferred. In order to retain some of the spatial locality, space-filling curves can be used to sample the data [12]. There are a number of different space-filling curves, but this work utilizes the Hilbert curve. Hilbert curves can be used to generate a one-dimensional signal that visits every point in two-dimensional space. Figure 2 shows the Hilbert curves of first, second and third order. These curves can be used to sample a matrix of size 4, 16, and 81, respectively. Higher-order curves can be used to sample larger input matrices, or the input data can be scaled to fit accordingly.

Corneal Topography

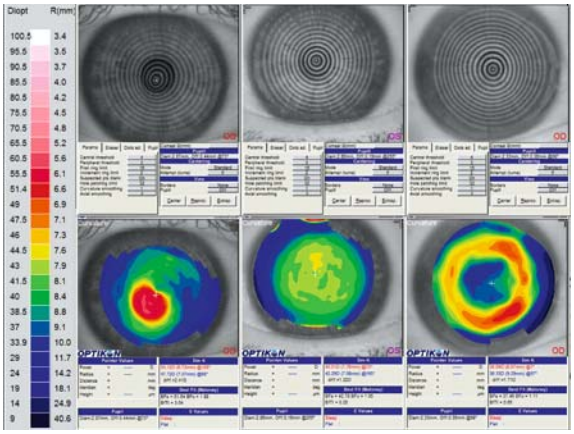
Clear vision depends on the optical quality of the corneal surface, which is responsible for nearly 75% of the total optical power of the eye [6]. Subtle distortions in the shape of the cornea can have a dramatic effect on vision. The process of mapping the surface features on the cornea is known as *corneal topography* [9].

The use of corneal topography has rapidly increased in recent years because of the popularity of refractive surgical procedures such as laser assisted in-situ keratomileusis (LASIK). Corneal topography is used to screen patients for corneal disease prior to this surgery and to monitor the effects of treatment after. It is also used to diagnose and manage certain diseases of the eye, including keratoconus, a progressive, non-inflammatory corneal disease that can lead to corneal transplant.

The most common method of determining corneal shape is to record an image of a series of concentric rings reflected from the corneal surface. Any distortion in corneal shape will cause a distortion of the concentric rings. By comparing the size and shape of the imaged rings with their



Biomedical Data Mining, Spatial, Figure 2 Graphical representation of First (**a**), Second (**b**), and Third-Order Hilbert Curves (**c**). Curves used to sample matrices of size 4, 16, and 81, respectively



Biomedical Data Mining, Spatial, Figure 3 Characteristic corneal shapes for three patient groups. The top image shows a picture of the cornea and reflected concentric rings. The bottom image shows the false color topographical map representing corneal curvature, with an increased curvature given a color in the red spectrum, decreased curvature in blue. From left to right: Keratoconus, Normal and post-refractive surgery

known dimensions, it is possible to mathematically derive the topography of the corneal surface.

Figure 3 shows an example of the output produced by a corneal topographer for a patient suffering from Keratoconus, a patient with a normal cornea, and an individual who has undergone LASIK eye surgery. The methods presented here are intended to differentiate between these patient groups. These images represent *illustrative* examples of each class, i. e. they are designed to be easily distinguishable by simple visual inspection. The top portion of the figure shows the imaged concentric rings. The bottom portion of the image shows a false color map representing the surface curvature of the cornea. This color map is intended to aid clinicians and is largely instrument-dependent.

Key Applications

This section covers techniques employed to represent the topography data with the above modeling methods, to use those representations to classify corneal shape, and finally, to present the results of those decisions to allow clinicians to visually inspect the classification criteria.

Data Transformation

The data from a corneal topographer are largely instrument-specific but typically consist of a 3D point cloud of approximately 7000 spatial coordinates arrayed in a polar grid. The height of each point z is specified by the relation $z = f(\rho, \theta)$, where the height relative to the corneal apex is a function of radial distance from the origin (ρ) and the counter-clockwise angular deviation from the horizontal meridian (θ). The inner and outer borders of each concentric ring consist of a discrete set of 256 data points taken at a known angle θ, but a variable distance ρ, from the origin.

Zernike The method described here is based on techniques detailed by Schwiegerling et al. and Iskander et al. [4,13]. In summary, the data is modeled over a user-selected circular region of variable-diameter, centered on the axis of measurement. The generation of the Zernike model surface proceeds in an iterative fashion, computing a point-by-point representation of the original data at each radial and angular location up to a user-specified limit of polynomial complexity. The polynomial coefficients of the surface that will later be used to represent the proportional magnitude of specific geometric features are computed by performing a least-squares fit of the model to the original data, using standard matrix inversion methods [13].

2D Wavelets To create a model using the 2D wavelet decomposition, the data must first be transformed from polar to Cartesian coordinates. Once this step has been completed, the matrix is normalized to a power of 2, typically 64x64 or 128x128. After normalization, the 2D wavelet decomposition is applied, with the final level of approximation coefficients serving as the feature vector.

1D Wavelets This section details two methods for transforming the topography data into a 1D signal. The first method is to simply trace along each ring in a counterclockwise fashion, adding each of the 256 points to the end of the signal. Upon reaching the end of a ring, one moves to the next larger ring and repeats the process. Next, a 1D decomposition is applied and the final level approximation coefficients are taken to serve as a feature vector for classification. The second method involves transforming the data into a distance matrix as with the 2D wavelet decomposition. Then, one can sample the data using a space-filling curve, which will result in a 1D representation of the 2D matrix.

Classification

Given a dataset of normal, diseased, and post-operative LASIK corneas, the above representations were tested using a number of different classification strategies, including decision trees, Naïve Bayes, and neural networks. The data was modeled with Zernike polynomials and several polynomials orders were tested, ranging from 4 to 10. The experiments showed that the low-order Zernike polynomials, coupled with decision trees, provided the best classification performance, yielding an accuracy of roughly 90% [11,16]. The 2D, 1D Hilbert and 1D ring-based wavelet representations were tested as well. For the 2D representation, a normalized matrix of size 128x128 and a 3rd level decomposition yielded the highest accuracy. For the Hilbert-based approach, a normalized matrix of size 64x64 and a 6th level transformation. With the ring-based model, the 7th level decomposition yielded the highest accuracy. The accuracy of the different wavelet-based models was roughly the same, hovering around 80%, approximately 10% lower than the best Zernike-based approach.

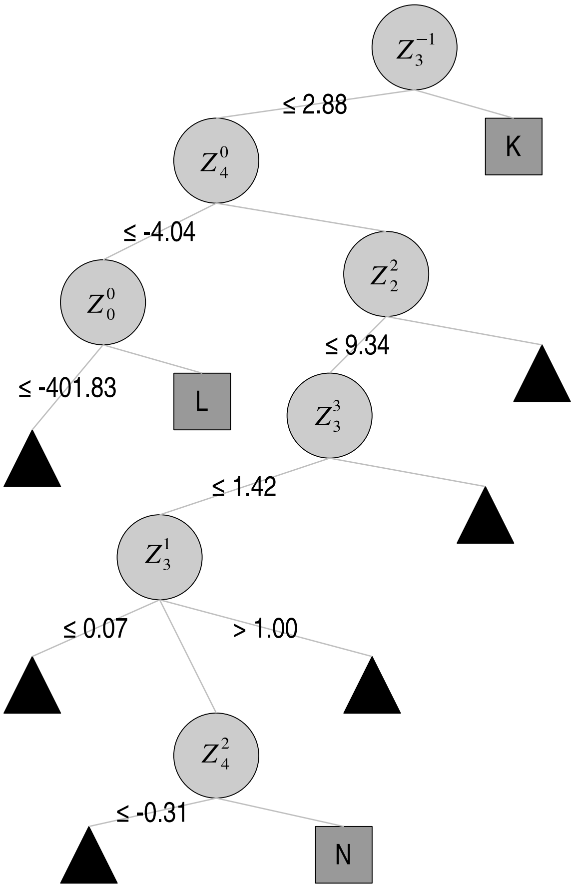
Visualization

While accuracy is an important factor in choosing a classification strategy, another attribute that should not be ignored is the interpretability of the final results. Classifiers like decision trees are often favored over “black-box” classifiers such as neural networks because they provide more understandable results. One can manually inspect the tree produced by the classifier to examine the features used in classification. In medical image interpretation, decision support for a domain expert is preferable to an automated classification made by an expert system. While it is important for a system to provide a decision, it is often equally important for clinicians to know the basis for an assignment.

Part of the rationale behind using Zernike polynomials as a transformation method over other alternatives is that there is a direct correlation between the geometric modes of the polynomials and the surface features of the cornea. An added benefit is that *the orthogonality of the series allows each term to be considered independently.* Since the polynomial coefficients used as splitting attributes represent the proportional contributions of specific geometric modes, one can create a surface representation that reflects the spatial features deemed “important” in classification. These features discriminate between the patient classes and give an indication as to the specific reasons for a decision.

The section below discusses a method that has been designed and developed to visualize decision tree results and aid in clinical decision support [10]. The first step of the process is to partition the dataset based on the path taken through the decision tree. Next a surface is created using the mean values of the polynomial coefficients of all the patients falling into each partition. For each patient, an individual polynomial surface is created from the patient’s coefficient values that correspond to the splitting attributes of the decision tree. This surface is contrasted against a similar surface that contains the mean partition values for the same splitting coefficients, providing a measure to quantify how “close” a patient lies to the mean.

Figure 4 shows a partial decision tree for a 4th Zernike model. The circles correspond to the Zernike coefficients used as splitting criteria, while the squares represent the leaf nodes. In this tree, three example leaf nodes are shown, one for each of the patient classes considered in this experiment (*K* - keratoconus, *N* - normal, *L* - post-operative LASIK). The black triangles are simply meant to represent subtrees that were omitted from the figure in order to improve readability. While each node of the tree represents one of the Zernike modes, the numbers on each branch represent the relation that must be true in order to proceed



Biomedical Data Mining, Spatial, Figure 4 Partial decision tree for a 4th order Zernike model

down that path. (If no relation is provided, it is simply the negation of the relation(s) on the opposing branch(es)). An object can be traced through the tree until a leaf node is reached and the object is then assigned the label of that leaf. Thus, given the tree in Fig. 4, if an object had a Z_3^{-1} coefficient with a value ≤ 2.88, one would proceed down the left branch. If the value was > 2.88, the right branch would be taken and the object would be labeled as keratoconus. In this manner, there is a path from the root node to each leaf.

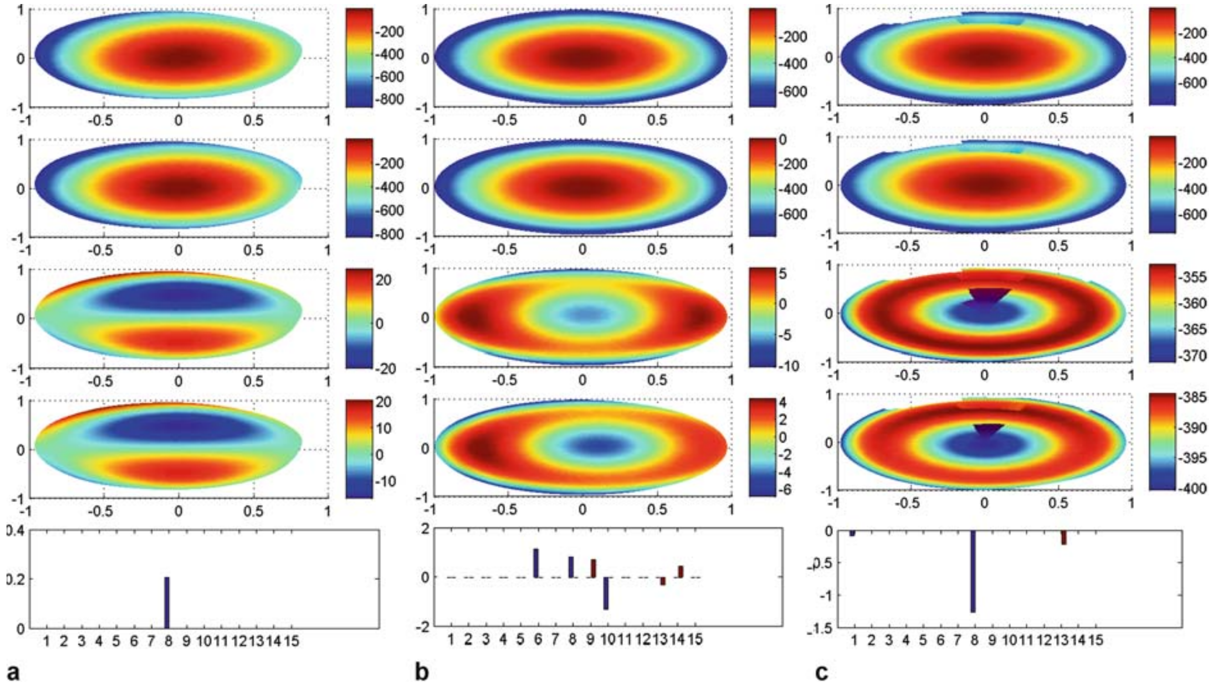
As a result, one can treat each possible path through a decision tree as a rule for classifying an object. For a given dataset, a certain number of patients will be classified by each rule. These patients will share similar surface features. Thus, one can compare a patient against the mean attribute values of all the other patients who were classified using the same rule. This comparison will give clinicians some indication of how "close" a patient is to the rule average. To compute these "rule mean" coefficients, denoted *rule*, the training data is partitioned and the average of each coefficient is calculated using all the records in that particular partition.

For a new patient, a Zernike transformation is computed and the record is classified using the decision tree to determine the rule for that patient. Once this step has been completed, the visualization algorithm is applied to produce five separate images (illustrated in Fig. 5). The first panel is a topographical surface representing the Zernike model for the patient. It is constructed by plotting the 3-D transformation surface as a 2-D topographical map, with elevation denoted by color. The second section contains a topographical surface created in a similar manner by using the *rule* coefficients. These surfaces are intended to give an overall picture of how the patient's cornea compares to the average cornea of all similarly-classified patients.

The next two panels in Fig. 5 (rows 3 and 4) are intended to highlight the features used in classification, i. e. the distinguishing surface details. These surfaces are denoted as *rule surfaces*. They are constructed from the value of the coefficients that were part of the classification rule (the rest are zero). The first rule surface (third panel of Fig. 5) is created by using the relevant coefficients, but instead of the patient-specific values, the values of the *rule* coefficients are used. This surface will represent the mean values of the distinguishing features for that rule. The second rule surface (row 4, Fig. 5) is created in the same fashion, but with the coefficient values from the patient transformation, not the average values.

Finally, a measure is provided to illustrate how close a patient lies to those falling in the same rule partition. For each of the distinguishing coefficients, the relative error between the patient and the *rule* is computed. The absolute value of the difference between the coefficient value of the patient and the value of the *rule* is taken and divided by the *rule* value. A bar chart of these error values is provided for each coefficient (the error values of the the coefficients not used in classification are set to zero). This plot is intended to provide a clinician with an idea of the influence of the specific geometric modes in classification and the degree that the patient deviates from the mean.

The surfaces in Fig. 5 correspond to the three example rules shown in the partial tree found in Fig. 4. For each of the patient classes, the majority of the objects were classified by the example rule. Since these are the most discriminating rules for each class, one would expect that the *rule surfaces* would exhibit surface features commonly associated with corneas of that type. These results are in agreement with expectations of domain experts.



Biomedical Data Mining, Spatial, Figure 5 Example surfaces for each patient class. **a** represents a Keratoconic eye, **b** a Normal cornea, and a post-operative LASIK eye. The top panel contains the Zernike representation of the patient. The next panel illustrates the Zernike representation using the *rule* values. The third and fourth panels show the *rule surfaces*, using the patient and *rule* coefficients, respectively. The bottom panel consists of a bar chart showing the deviation between the patient's *rule surface* coefficients and the *rule* values

Future Directions

The above methods are not limited just to modeling the shape of the cornea. They could be applied to almost any domain dealing with structure-based data.

Cross References

► Space-Filling Curves

Recommended Reading

1. Born, M.,Wolf E.: Principles of optics: electromagnetic theory of propagation, interference and diffraction of light, 6th edn. Pergamon Press, Oxford, New York (1980)
2. Daubechies, I.: Ten Lectures on Wavelets. Soc. Indust. Appl. Math., Philadelphia (1992)
3. Hoekman, D.H., Varekamp, C.: Observation of tropical rain forest trees by airborne high-resolution interferometric radar. Geosci. Remote Sensing, IEEE Trans on, **39**(3),584–594 (2001)
4. Iskander, D.R., Collins, M.J., Davis, B.: Optimal modeling of corneal surfaces with zernike polynomials. IEEE Trans Biomed. Eng. **48**(1),87–95 (2001)
5. Iskander, D.R., Collins, M.J., Davis, B., Franklin, R.: Corneal surface characterization: How many zernike terms should be used? (ARVO) abstract. Invest. Ophthalmol. Vis. Sci. **42**(4),896 (2001)
6. Kiely, P.M., Smith, G., Carney, L.G.: The mean shape of the human cornea. J. Modern Opt. **29**(8),1027–1040 (1982)
7. Laine, A.F.: Wavelets in temporal and spatial processing of biomedical images. Annu. Rev. Biomed. Eng. **02**, 511–550 (2000)
8. Mallat, S.: A Wavelet Tour of Signal Processing, 2nd edn. Academic, New York (1999)
9. Mandell, R.B.: A guide to videokeratography. Int. Contact Lens Clin. **23**(6),205–28 (1996)
10. Marsolo, K., Parthasarathy, S., Twa, M.D., Bullimore, M.A.: A model-based approach to visualizing classification decisions for patient diagnosis. In: Proceedings of the Conference on Artificial Intelligence in Medicine (AIME), Aberdeen, 23–27 July 2005
11. Marsolo, K., Twa, M., Bullimore, M.A., Parthasarathy, S.: Spatial modeling and classification of corneal shape. IEEE Transactions on Information Technology in Biomedicine (2006)
12. Platzman, L., Bartholdi, J.: Spacefilling curves and the planar travelling salesman problem. J. Assoc. Comput. Mach. **46**, 719–737 (1989)
13. Schwiegerling, J., Greinvenkamp, J.E., Miller, J.M.: Representation of videokeratoscopic height data with zernike polynomials. J. Opt. Soc. Am. A **12**(10),2105–2113 (1995)
14. Teh, C.-H., Chin, R.T.: On image analysis by the methods of moments. IEEE Trans. on Pattern. Anal. Mach. Intell. **10**(4):496–513 (1988)
15. Thibos, L.N., Applegate, R.A., Schwiegerling, J.T., Webb, R.: Standards for reporting the optical aberrations of eyes. In: TOPS, Santa Fe, NM (1999) OSA
16. Twa, M.D., Parthasarathy, S., Raasch, T.W., Bullimore, M.A.: Automated classification of keratoconus: A case study in analyzing clinical data. In: SIAM Int'l Conference on Data Mining, San Francisco, CA, 1–3 May 2003

17. Zernike, F.: Beugungstheorie des Schneidenverfahrens und seiner verbesserten Form, der Phasenkontrastmethode. Physica, **1**, 689–704 (1934)

Bitmap

► Raster Data

BLUP

► Spatial Econometric Models, Prediction

Branch and Bound

► Skyline Queries

B^x-Tree

► Indexing of Moving Objects, B^x-Tree

B-Tree, Versioned

► Smallworld Software Suite

Bundle Adjustment

► Photogrammetric Methods

Business Application

► Decision-Making Effectiveness with GIS

Caching

► Olap Results, Distributed Caching

CAD and GIS Platforms

► Computer Environments for GIS and CAD

Cadaster

► Cadastre

Cadastre

Erik Stubkjær
Department of Development and Planning,
Aalborg University, Aalborg, Denmark

Synonyms

Cadaster; Land administration system; Land information system; Land registry; Property register; Land policy; Spatial reference frames

Definition

A cadastre may be defined as an official geographic information system (GIS) which identifies geographical objects within a country, or more precisely, within a jurisdiction. Just like a land registry, it records attributes concerning pieces of land, but while the recordings of a land registry is based on deeds of conveyance and other rights in land, the cadastre is based on measurements and other renderings of the location, size, and value of units of property. The cadastre and the land registry in some countries, e. g., the Netherlands and New Zealand, are managed within the same governmental organization. From the 1990s, the term "land administration system" came into use, referring to a vision of a complete and consistent national information system, comprising the cadastre and the land registry.

The above definition of cadastre accommodates to the various practices within continental Europe, the British Commonwealth, and elsewhere. Scientific scrutiny emerged from the 1970s, where the notions of a land information system or property register provided a frame for comparison of cadastres across countries. However, GIS and sciences emerged as the main approach of research in the more technical aspects of the cadastre. In fact, the above definitional exercise largely disregards the organizational, legal, and other social science aspects of the cadastre. These are more adequately addressed when the cadastre is conceived of as a sociotechnical system, comprising technical, intentional, and social elements.

Historical Background

The notion of the cadastre has been related to Byzantine ledgers, called *katastichon* in Greek, literally "line by line". A Roman law of 111 BC required that land surveyors (*agrimensores*) should complete maps and registers of certain tracts of Italy. Also, an archive, a *tabularium*, was established in Rome for the deposit of the documents. Unfortunately, no remains of the *tabularium* seem to have survived the end of the Roman Empire.

The Cadastre reemerged in the fifteenth century in some Italian principalities as a means of recording tax liabilities. This seems part of a more general trend of systematic recording of assets and liabilities, e. g., through double-entry bookkeeping, and spread to other parts of Europe. In order to compensate for the lack of mapping skills, landed assets and their boundaries were described through a kind of written maps or *cartes parlantes*. During the sixteenth century, landscape paintings or so called "picture maps" were prepared e. g., for the court in Speyer, Germany, for clarifying argumentation on disputed land. During the same period in the Netherlands, the need for dike protection from the sea called for measurements and the organization of work and society; the practice of commissioning surveys for tax collection became increasingly common there.

A new phase was marked by new technology, the plane table with the related methods for distance measurement, mapping, and area calculation. The technology was introduced in 1720 in Austrian Lombardy through a formal trial against alternative mapping methods. The resulting Milanese census, the *Censimento*, with its integrated recordings in ledgers and maps became a model for other European principalities and kingdoms.

The uneven diffusion of cadastral technology reveals a power struggle between the ruling elite and the landed gentry and clerics, who insisted on their tax exemption privileges. In the early modern states, the cadastre was motivated by reference to a "God-given principle of equality" (German: *gottgefällige Gerechtigkeit'* or *gottgefällige Gleichheit* [1]). Generally, absolutist monarchs were eager to establish accounts of the assets of their realm, as the basis for decisions concerning their use in wars and for the general benefit of the realm. A continental European version of mercantilism, "cameralism", was lectured at universities, seeking a quasirational exploitation of assets and fair taxation, for which the cadastre was needed, as well as regulations and educational programs, for example in agriculture, forestry, and mining. Cadastral technology, the related professions, and the centralized administration together became an instrument of unification of the country, providing the technical rationale for greater equality in taxation. Taxation thus gradually became controlled by the central authority, rather than mediated through local magnates. This change was recognized by Adam Smith, by physiocrats, and by political writers in France. The administrative technology was complemented by codification, that is, a systematic rewriting of laws and by-laws that paved the way for the modern state where individual citizens are facing the state, basically on equal terms.

The reorganization of institutions after the French revolution of 1789 also changed the role of the cadastre as introduced by Enlightenment monarchs. In part, this was due to university reforms, e. g., by Wilhelm von Humboldt in Berlin. Cameralism was split into economics, which increasingly became a mathematically based discipline and a variety of disciplines lectured at agricultural and technical universities. Cadastral expertise was largely considered a subfield of geodesy, the new and rational way of measuring and recording the surface of the Earth. From the end of eighteenth century, cadastral maps were increasingly related to or based on geodetic triangulations, as was the case for the Napoleonic cadastre of France, Belgium and the Netherlands.

The same cadastral reform included the intention of using the measured boundaries and areas and the parcel identification for legal purposes, primarily by letting the cadastral documentation with its "fixed boundaries" prove title to land and become the final arbiter in case of boundary disputes. However, the courts that were in charge of land registries, generally for a more than a century, and the legal profession were reluctant to adopt what might be considered an encroachment on their professional territory. During the nineteenth century, most countries improved their deeds recording system, and German-speaking countries managed to develop them into title systems backed by legal guaranties. Similarly, from South Australia the so-called Torrens system, adopted in 1858, influenced the English-speaking world. However, with few exceptions, the integration of cadastral and legal affairs into one information system had to await the introduction of computer technology and the adoption of business approaches in government.

The above historical account is Eurocentric and bypasses possible scientific exchange with South-Eastern neighbors during the twelfth to fifteenth centuries. Also, it leaves out the development in England and its colonies worldwide. The account describes how the notion of the cadastre emerged and varied across time and place. This calls for special care in scientific communications, since no standardized and theory-based terminology has been established.

Scientific Fundamentals

Spatial Reference Frames

The center and rotational axis of the Earth, together with the Greenwich meridian, provide a reference frame for the location of objects on the surface of the Earth. Furthermore, a map projection relates the three-dimensional positions to coordinates on a two-dimensional map plane. The skills of the geodesist and land surveyor are applied in the cadastral field to record agreed legal boundaries between property units, the assumption being that such recordings substantially reduce the number of boundary disputes.

Application of the above-mentioned "fixed-boundary" solution raises serious problems, even if the assumption may be largely confirmed. In the cases of land slides and land drifting due to streams, the solution is insensitive to the owner's cost of getting access to all parts of the property. Furthermore, the solution does not accommodate for later and better measurements of the boundary. Moreover, the boundary may have been shifted years ago for reasons that have become too costly to elicit, relative to the value of the disputed area and even acknowledging the fact that justice may not be served. Some jurisdictions hence allow for "adverse possession", that is: an official recognition of neighbors' agreement on the present occupational boundary, even if it differs from previous cadastral recordings. Likewise, legal emphasis on merestones and other boundary marks, as well as the recording of terrain features which determine permanent and rather well defined boundary points may supplement the pure fixed-boundary approach.

The geodetic surveyors' reference frames locate points in terms of coordinates, but the naming and identification of parcels relates to names, which is a subfield of linguistics. The cadastral identifier is a technical place name, related to the place names of towns, roads, parishes, and topographic features. Hierarchically structured administrative units or jurisdictions and their names provide a means of location of property units. Even if such ordinal structuring of a jurisdiction through place names is coarse, relative to the metric precision of measured boundary points, it provides in many cases for a sufficient localization of economic activities and it reduces the dependency of specialized and costly competence.

The linguistic approach to localization refers to another spatial reference frame than the Earth, namely the human body. The everyday expressions of "left" and "right, "up" and "down" all refer to the body of the speaker or the listener, as practiced when giving directions to tourists or explaining where to buy the best offers of the day. Important research areas include the balancing of nominal, ordinal and metric means of localization and the consideration of relations amongst various spatial reference frames.

Communication and Standardization

The national information systems (cadastre, land registry, or whatever may be the name) and organizational structure of real property information systems depend on databases and related archives and need updating if they are to render trustworthy minutes of the spatial and legal situation of the property units. Computer and communication sciences provide the theoretical background for these structures and tasks. However, until recently the methods provided in terms of systems analysis and design, data modeling, etc. addressed the information system within a single decision body, while the situation pertaining to real property information depends on an interplay between ministries, local government, and the private sector. The notion of a geospatial data infrastructure is used to describe this scope. The modeling of this infrastructure compares to the modeling of an industrial sector for e-business, an emergent research issue that includes the development of vocabulary and ontology resources of the domain.

The specification of the property unit is a fundamental issue. Often, the unit is supposed to be in individual ownership, but state or other public ownership is common enough to deserve consideration, as are various forms of collective ownership. Collective ownership is realized by attributing rights and obligations to a so-called legal person, which may be an association, a limited company, or another social construct endorsed by legislation or custom. Comparative studies reveal that the property unit itself can be specified in a host of variations: Is the unit a single continuous piece of land, or is it defined as made up of one or more of such entities? Relations among pieces of land can be specified in other ways: In Northern and Central Europe, a construct exists where a a certain share of a property unit is owned by the current owners of a number of other property units. In cadastral parlance, such unit is said to be owned by the share-holding property units. Furthermore, land and the building erected on it may establish one unit, yet alternatively the building always or under certain conditions constitutes a unit in its own right. Variations also occur as to whether parts of buildings can become units, for example in terms of condominiums, which may depend on conditions related to use for housing or business purposes. Research efforts under the heading of "standardization of the core cadastral unit" have contributed substantially to the understanding of the complexity of the property unit.

Updating the information in property registers is as essential as the specification of units. From an informatics point of view, a survey of the information flows in what may be called the "geodata network" may reveal uncoordinated, and perhaps duplicated, efforts to acquire information and other suboptimal practices. However, from the end users' point of view, what takes place is a transaction of property rights and related processes, for example subdivision of property units. The updating of property registers is from this point of view a by-product of the transaction.

The end-user point of view is taken also by economists, who offer a theoretical basis for investigations of the mentioned processes, a field known as "institutional economics". New institutional economics (NIE) introduces "transaction costs" as an expense in addition to the cost of producing a commodity to the market. In the present context, the transaction costs are the fees and honoraries, etc. to be paid by the buyer and seller of a property unit, besides the cost of the property itself. Buyers' efforts to make sure that the seller is in fact entitled to dispose of the property unit concerned can be drastically reduced, that is: transaction costs are lowered, where reliable title information from land registries is available.

The NIE approach, as advocated by Nobel laureate Douglass C North, was applied in recent, comparative research. His notion of "institution": the norms which restrict and enable human behavior, suggested research focused on practices rather than legal texts. Methods were developed and applied for the systematic description and comparison of property transactions, including a formal, ontology-based approach. The methods developed were feasible and initial national accounts of transaction costs were drafted. However, advice for optimal solutions are not to be expected, partly because many of the agents involved, both in the private and the public sector, have other preferences than the minimizing of transaction costs. Moreover, property transactions are, for various political reasons, often regulated, for example through municipal preemption rights or spatial planning measures. The NIE approach does however offer an adequate theoretical basis for analyzes of the infrastructure of real property rights, analyzes which assist in the identification and remedy of the most inappropriate practices.

Cadastral Development Through Institutional Transactions

Institutional economics divides into two strands: NIE, and institutional political economy, respectively. The former may be applied to the cadastre and its related processes, conceived as a quasirational, smooth-running machine that dispatches information packets between agents to achieve a certain outcome, e. g., the exchange of money against title to a specific property unit. However, this approach does not account for the fact that the various governmental units, professional associations, etc. involved in the cadastral processes have diverse mandates and objectives. Development projects pay attention to these conflicting interests through stakeholder analyzes. Research in organizational sociology suggests identification of "policy issue networks" and investigation of the exchange of resources among the actors involved, for example during the preparation and passing of a new act. The resources are generally supposed not to be money, although bribery occurs, but more abstract entities such as legal–technical knowledge, access to decision centers, organizational skills, reputation, and mobilizing power.

Institutional economics provides a frame for relating this power game to the routine transactions of conveyance of real estate. It is done by introducing two layers of social analysis: the layer of routine transactions, and the layer of change of the rules which determine the routine transactions [2]. In economic terms, conveyance is an example of a transaction in commodities or other assets. These transactions are performed according to a set of rules: acts, by-laws, professional codes of conduct, etc. which are, in the terminology of Douglass North, a set of institutions. The process of change of these institutions is the object of analysis on the second layer and, following Daniel Bromley, called "institutional transactions". Institutional transactions may reduce transaction costs within the jurisdiction concerned, but Bromley shows at length that this need not be so; generally, the initiator of an institutional transaction cannot be sure whether the intended outcome is realized. Among other things, this is because the transaction is open to unplanned interference from actors on the periphery and also because the various resources of the actors are only partly known at the outset.

The strand of institutional political economy is researching such institutional transactions in order to explain why some countries grow rich, while others fail to develop their economy. Here we have the theoretical basis for explaining the emergence and diffusion of the cadastre in early modern Europe, cf. "Historical Background" above. The pious and enlightened absolutist monarchs and their advisors established a set of norms that framed institutional transactions in a way that encouraged a growing number of the population to strive for "the common weal".

Key Applications

The definition of cadastre specifies the key application: the official identification and recording of information on geographical objects: pieces of land, buildings, pipes, etc. as well as documents and extracts from these on rights and restrictions pertaining to the geographical objects. Cadastral knowledge is applied not only in the public sector, but also in private companies, as we shall see form the following overview of services:

- Facilitation of financial inflow to central and local government by specification of property units and provision of information on identification, size, and boundaries and by providing standardized assessment of property units (mass-appraisal)
- Supporting the market in real estate by providing trustworthy recordings of title, mortgages, and restrictions
- Supporting sustainable land use (urban, rural, and environmental aspects) by providing data for the decision process and applying the decision outcome in property development
- Supporting the construction industry and utilities with data and related services, and cooperating on the provision of statistics
- Assisting in continuous improvement of administration, often called "good governance", through:
  - Considering data definitions and data flows with other bodies in government and industry, by mutual relating of post addresses, units of property, units for small-area statistics and spatial planning, as well as road segments and other topographic elements
  - Applying cost-effective technology
  - Contributing towards good governance through participating in national high-level commissions that aim at change in the institutional structure concerning real property rights, housing, and land use.

The taxation issue is mentioned first among public sector issues, partly for historical reasons, but more essentially because the operation of the cadastre cost money. In a specific country, the list of tasks or functions may include more or less operations, and the grouping of tasks may differ. However, the list may give an idea of career opportunities, as does the corresponding list of services within the private sector:

- Assisting bodies in need of recording and management of a property portfolio, for example utilities, road, railroad and harbor agencies, defense, property owners' associations and hunting societies, charitable trusts, ecclesiastical property; supporting the selection of antenna sites for wireless networks
- Supporting property development by assisting owners in reorganizing the shape of and rights in the property unit and its surroundings, in compliance with the existing private and public restrictions, for example easements and zoning. Delivering impartial expertise in boundary disputes.
- Assisting bodies in need of measuring and recording of location-specific themes with legal implications, including cultural and natural heritage and opencast mining
- Facilitate the diffusion of open source and proprietary GIS software by adapting it to local needs

Again, in a specific country the structure of the construction industry and the utilities, as well as the status and scope of professional associations like those of geodetic surveyors and property developers may vary.

Future Directions

Formalization of Practice

Much of cadastral knowledge is still tacit. Thus a large task ahead is to elicit this knowledge and integrate it with relevant existing conceptual structures and ontologies. Cadastral work is embedded in legal prescripts and technical standards. The hypothesis supported by institutional economics is that the needed formalization will be achieved better through observation and analysis of human routine behavior than through legal analysis of prescripts or model building of information systems. Furthermore, the need for formalization is not only in regard to the production of cadastral services, but also the articulation of the intricate structure of process objectives or functions. The description of realized functions and function clusters depends to a certain extent on the local expertise that has an obvious interest in staying in business. The description effort thus has to face a value-laden context. Good places to begin research are with Kuhn [3], Frank [4,5], Oosterom et al. [6], and Zevenbergen et al. [7].

Prioritized Issues

The International Federation of Surveyors (FIG) provides a framework for exchanges among professionals and academics. Of the ten commissions, commission 7 on Cadastre and Land Management at the International FIG Congress in Munich, October 2006, adopted a Work Plan 2007–2010. Major points of the plan include:

- Pro-poor land management and land administration, supporting good governance
- Land administration in post conflict areas
- Land administration and land management in the marine environment
- Innovative technology and ICT support in land administration and land management
- Institutional aspects for land administration and land management

Current working relations with UN bodies will continue and possibly include the UN Habitat Global Land Tool Network (GLTN). A good starting point would be http://www.fig.net/commission7/ and Kaufmann and Steudler [8].

Development Research

From a global perspective, the past decades have been marked by an attempt, on the part of donors and experts, to install the Western model of individual real property rights and cadastre in developing or partner countries. The efforts have not had the intended effects for a variety of reasons. One may be that cadastral processes interfere with the preferences and habits of end users, including the large number of owners, lease-holders, tenants, mortgagees, etc. This is the case especially where the general economy and the structure of professions and university departments hamper the provision of skilled and impartial staff (in public and private sector) who could mediate between the wishes of the right holders and the technicalities of property rights, transactions, and recordings.

The provision of resources for such cadastral development likely depends on one or more global networks which comprises university departments, central and local governments, nongovernmental organizations (NGOs) and international organizations, for example as demonstrated in the Cities Alliance. Furthermore, the United Nations University (UNU) includes UNU-IIST, the International Institute for Software Technology, enabling software technologies for development, as well as UNU-WIDER which analyses property rights and registration from the point of view of development economics.

The Netherlands-based International Institute for Geo-Information Science and Earth Observation (ITC) has become an associated institution of UNU. With the Netherlands Cadastre, Land Registry and Mapping Agency, the ITC is establishing a School for Land Administration Studies at ITC. The school will among others execute a joint land administration programme with UNU, consisting of a series of seminars, short courses and networking. Bruce [9], Janvry and Sadoulet [10]; Palacio [11], and North [12] are good places to start further reading on this.

Spatial Learning: Naming Objects of the Environment

In autumn 2006, the US National Science Foundation awarded a $3.5 million grant to establish a new research center to investigate spatial learning and use this knowledge to enhance the related skills students will need. Research goals include how to measure spatial learning.

Cadastral studies combine the verbal and the graphical–visual-spatial modes of communication and need reflection of the teaching and learning methods, also with web-supported distance learning in mind. More specifically, the need of balancing of nominal, ordinal and metric means of localization, etc. which was mentioned above in "Spatial Reference Frames", would benefit from such reflection.

Recommended Reading

1. Kain, R.J.P., Baigent, E.: The Cadastral Map in the Service of the State: A History of Property Mapping. The University of Chicago Press, Chicago (1993)
2. Williamson, O.E.: The new institutional economics: taking stock, looking ahead. J. Econ. Literat. **38**, 595–613 (2000)
3. Kuhn, W.: Ontologies in support of activities in geographical space. Int. J. Geogr. Inf. Sci. **15**, 613–631 (2001)
4. Frank, A.U.: Tiers of ontology and consistency constraints in geographic information systems. Int. J. Geogr. Inf. Sci. **15**, 667–678 (2001)
5. Stubkjær, E.: Modelling units of real property rights. In: Virrantaus, K., Tveite, H. (eds.) ScanGIS'03 Proceedings, 9th Scandinavian Research Conference on Geographical Information Sciences, June 2003, Espoo, Finland (2003)
6. van Oosterom, P., Schlieder, C., Zevenbergen, J., Hess, C., Lemmen, C., Fendel, E. (eds.): Standardization in the cadastral domain. In: Proceedings, Standardization in the Cadastral Domain, Bamberg, December 2004 Frederiksberg. The International Federation of Surveyors (2005)
7. Zevenbergen, J., Frank, A., Stubkjær, E. (eds.:) Real Property Transactions: Procedures, Transaction Costs and Models. IOS Press, Amsterdam (2007)
8. Kaufmann, J., Steudler, D. (1998) Cadastre 2014. http://www.fig.net/cadastre2014/index.htm. Accessed 14 Aug 2007
9. Bruce, J. (1993) Review of tenure terminology. Land Tenure Center Report. http://agecon.lib.umn.edu/ltc/ltctb01.pdf. Accessed 14 Aug 2007
10. de Janvry, A., Sadoulet, E. (2001) Access to Land and Land Policy Reforms. Policy Brief No. 3, April 2001. http://www.wider.unu.edu/publications/pb3.pdf. Accessed 14 Aug 2007
11. Palacio, A.: Legal Empowerment of the Poor: An Action Agenda for the World Bank. Available via http://rru.worldbank.org/Documents/PSDForum/2006/background/legal_empowerment_of_poor.pdf. Accessed 14 Aug 2007
12. North, D.C.: Institutions, Institutional Change and Economic Performance. Cambridge University Press, Cambridge, UK (1990)
13. Chang, H.-J.: Understanding the Relationship between Institutions and Economic Development: Some Key Theoretical Issues. UNU/WIDER Discussion Paper 2006/05. July 2006. http://www.wider.unu.edu/publications/dps/dps2006/dp2006-05.pdf. Accessed 14 Aug 2007
14. de Janvry, A., Gordillo, G., Platteau, J.-P., Sadoulet, E. (eds.): Access to Land, Rural Poverty and Public Action. UNU/WIDER Studies in Development Economics. Oxford University Press, UK and Europe (2001)

Camera Model

► Photogrammetric Methods

Cartographic Data

► Photogrammetric Products

Cartographic Generalization

► Abstraction of GeoDatabases

Cartographic Information System

► Multimedia Atlas Information Systems

Catalog Entry

► Metadata and Interoperability, Geospatial

Catalogue Information Model

Liping Di, Yuqi Bai
Center for Spatial Information Science and Systems (CSISS), George Mason University, Greenbelt, MD, USA

Synonyms

Catalogue metadata schema; Catalogue information schema; Registry information model

Definition

The catalogue information model is a conceptual model that specifies how metadata is organized within the catalogue. It defines a formal structure representing catalogued resources and their interrelationships, thereby providing a logical schema for browsing and searching the contents in a catalogue.

There are multiple and slightly different definitions of the catalogue information model used by various communities. The Open Geospatial Consortium (OGC) defines the catalogue information model in the OGC Catalogue Services Specification [1] as an abstract information model that specifies a BNF grammar for a minimal query language, a set of core queryable attributes (names, definitions, conceptual data types), and a common record format that defines the minimal set of elements that should be returned in the brief and summary element sets. The Organization for the Advancement of Structured Information Standards (OASIS) defines the registry information model in the ebXML Registry Information Model (ebRIM) specification [2] as the information model which provides a blueprint or high-level schema for the ebXML registry. It provides the implementers of the ebXML registry with information on the type of metadata that is stored in the registry as well as the relationships among metadata classes. The registry information model defines what types of objects are stored in the registry and how stored objects are organized in the registry. The common part from these two definitions is the schema for describing the objects cataloged/registered in and for organizing the descriptions in a catalogue/registry – the catalogue metadata schema.

Historical Background

The first catalogues were introduced by publishers serving their own business of selling the books they printed. At the end of the fifteenth century, they made lists of the available titles and distributed them to those who frequented the book markets. Later on, with the increasing volume of books and other inventories, the library became one of the earliest domains providing a detailed catalogue to serve their users. These library catalogues hold much of the reference information (e. g., author, title, subject, publication date, etc.) of bibliographic items found in a particular library or a group of libraries.

People began to use the term "metadata" in the late sixties and early seventies to identify this kind of reference information. The term "meta" comes from a Greek word that denotes "alongside, with, after, next." More recent Latin and English usage would employ "meta" to denote something transcendental or beyond nature [3].

The card catalogue was a familiar sight to users for generations, but it has been effectively replaced by the computerized online catalogue which provides more advanced information tools helping to collect, register, browse, and search digitized metadata information.

Scientific Fundamentals

Metadata can be thought of as data about other data. It is generally used to describe the characteristics of information-bearing entities to aid in the identification, discovery, assessment, management, and utilization of the described entities. Metadata standards have been developed to standardize the description of information-bearing entities for specific disciplines or communities. For interoperability and sharing purposes, a catalogue system usually adopts a metadata standard used in the community the system intends to serve as its catalogue information model.

A metadata record in a catalogue system consists of a set of attributes or elements necessary to describe the resource in question. It is an example of the catalogue information model being used by the catalogue system. A library catalogue, for example, usually consists of the author, title, date of creation or publication, subject coverage, and the call number specifying the location of the item on the shelf. The structures, relationships, and definitions for these queryable attributes – known as conceptual schemas – exist for multiple information communities. For the purposes of interchange of information within an information community, a metadata schema may be created that provides a common vocabulary which supports search, retrieval, display, and association between the description and the object being described.

A catalogue system needs to reference an information model for collecting and manipulating the metadata of the referenced entities cataloged in the system. The information model provides specific ways for users to browse and search them. Besides the metadata information that directly describes those referenced entities themselves, a catalogue might hold another type of metadata information that describes the relationship between these entities.

Some catalogue services may only support one catalogue information model, each with the conceptual schema clearly defined, while others can support more than one catalogue information model. For example, in the US Geospatial Data Clearinghouse, the affiliated Z39.50 catalogue servers only support US Content Standard for Digital Geospatial Metadata (CSDGM) standard in their initial developing stage. While in OGC Catalogue Service base specification, what catalogue information model can be used is undefined. Developers are encouraged to propose their own catalogue information model as profiles.

However, to facilitate the interoperability between diverse OGC-compliant catalogue service instances, a set of core queryable parameters originated from Dubin core is proposed in the base specification and is desirable to be supported in each catalogue service instance. OGC further endorsed the OASIS ebRIM (e-Business Registry Information Model) as the preferred basis for future profiles of OGC Catalogue [11].

How a catalogue information model can be formally discovered and described in a catalogue service is another issue. Some catalogue services do not provide specific operations for automatic discovery of the underlying catalogue information model, while others support particular operations to fulfill this task. In the OGC Catalogue Service specification, the names of supported Information Model Elements can be listed in the Capabilities files and a mandatory *DescribeRecord* operation allows the client to discover elements of the information model supported by the target catalogue service. This operation allows some of or the entire information model to be described.

Key Applications

The concept of the catalogue information model has been widely applied in many disciplines for information management and retrieval Common metadata standards are widely adopted as the catalogue information model. Among them, the Dublin Core is one of the most referenced and commonly used metadata information models for scientific catalogues. In the area of geographic information science, ISO 19115 is being widely adopted as the catalogue information model for facilitating the sharing of a large volume of geospatial datasets.

Dublin Core

The Dublin Core metadata standard is a simple yet effective element set for describing a wide range of networked resources [4]. The "Dublin" in the name refers to Dublin, Ohio, USA, where the work originated from a workshop hosted by the Online Computer Library Center (OCLC), a library consortium which is based there. The "Core" refers to the fact that the metadata element set is a basic but expandable "core" list [3].

The Simple Dublin Core Metadata Element Set (DCMES) consists of 15 metadata elements: Title, Creator, Subject, Description, Publisher, Contributor, Date, Type, Format, Identifier, Source, Language, Relation, Coverage and Rights. Each element is optional and may be repeated.

The Dublin Core Metadata Initiative (DCMI) continues the development of exemplary terms or "qualifiers" that extend or refine these original 15 elements. Currently, the DCMI recognizes two broad classes of qualifiers: Element Refinement and Encoding Scheme. Element Refinement makes the meaning of an element narrower or more specific. Encoding Scheme identifies schemes that aid in the interpretation of an element value.

There are many syntax choices for Dublin Core metadata, such as SGML, HTML, RDF/XML, and Key-Value Pair TXT file. In fact, the concepts and semantics of Dublin Core metadata are designed to be syntax independent and are equally applicable in a variety of contexts.

Earth Science

With the advances in sensor and platform technologies, the Earth science community has collected a huge volume of geospatial data in the past thirty years via remote sensing methods. To facilitate the archival, management, and sharing of these massive geospatial data, Earth science community has been one of the pioneers in defining metadata standards and using them as information models in building catalogue systems.

FGDC Content Standard for Digital Geospatial Metadata The Federal Geographic Data Committee (FGDC) of the United States is a pioneer in setting geospatial metadata standards for the US federal government. To provide a common set of terminology and definitions for documenting digital geospatial data, FGDC initiated work on setting the Content Standard for Digital Geospatial Metadata (CSDGM) in June of 1992 through a forum on geospatial metadata. The first version of the standard was approved on June 8, 1994, by the FGDC.

Since the issue of Executive Order 12906, "Coordinating Geographic Data Acquisition and Access: The National Spatial Data Infrastructure," by President William J. Clinton on April 11, 1994, this metadata standard has been adopted as the catalogue information model in numerous geospatial catalogue systems operated by US federal, state, and local agencies as well as companies and groups. It has also been used by other nations as they develop their own national metadata standards.

In June of 1998, the FGDC approved the CSDGM version 2, which is fully backward compatible with and supersedes the June 8, 1994 version. This version provides for the definition of Profiles (Appendix E) and extensibility through User Defined Metadata Extensions (Appendix D). The June 1998 version also modifies some production rules to ease implementation.

The Content Standard for Digital Geospatial Metadata (CSDGM) [5] identifies and defines the metadata elements used to document digital geospatial data sets for many purposes, which includes 1) preservation of the meaning and value of a data set; 2) contribution to a catalogue or clearinghouse and; and 3) aid in data transfer. CSDGM groups the metadata information into the following seven types:

- Identification_Information
- Data_Quality_Information
- Spatial_Data_Organization_Information
- Spatial_Reference_Information
- Entity_and_Attribute_Information
- Distribution_Information
- Metadata_Reference_Information

For each type, it further defines composed elements and their type, short name and/or Domain information.

To provide a common terminology and set of definitions for documenting geospatial data obtained by remote sensing, the FGDC defined the Extensions for Remote Sensing Metadata within the framework of the June 1998 version of the CSDGM [6]. These *Remote Sensing Extensions* provide additional information particularly relevant to remote sensing: the geometry of the measurement process, the properties of the measuring instrument, the processing of raw readings into geospatial information, and the distinction between metadata applicable to an entire collection of data and those applicable only to component parts. For that purpose, these *Remote Sensing Extensions* establish the names, definitions, and permissible values for new data elements and the compound elements of which they are the components. These new elements are placed within the structure of the base standard, allowing the combination of the original standard and the new extensions to be treated as a single entity, [6].

ISO 19115 In May 2003, ISO published ISO 19115: Geographic Information-Metadata [7]. The international standard was developed by ISO Technical Committee (TC) 211 as a result of consensus among TC national members as well as its liaison organizations on geospatial metadata. ISO 19115, rooted at FGDC CSDGM, provides a structure for describing digital geographic data. Actual clauses of 19115 cover properties of the metadata: identification, constraints, quality, maintenance, spatial representation (grid and vector), reference systems, content (feature catalogue and coverage), portrayal, distribution, extensions and application schemas. Complex data types used to describe these properties include extent and citations. ISO 19115 has been adopted by OGC as a catalogue information model in its Catalogue Service for Web-ISO 19115 Profile [8]. Figure 1 depicts the top level UML model of the metadata standard.

ISO 19115 defines more than 300 metadata elements (86 classes, 282 attributes, 56 relations). The complex, hierarchical nested structure and relationships between the components are shown using 16 UML diagrams.

To address the issue whether a metadata entity or metadata element shall always be documented in the metadata or sometimes be documented, ISO 19115 defines a descriptor for each package and each element. This descriptor may have the following values:

– M (mandatory)
– C (conditional) or
– O (optional).

Mandatory (M) means that the metadata entity or metadata element shall be documented.

Conditional (C) specifies an electronically manageable condition under which at least one metadata entity or a metadata element is mandatory. Conditions are defined in the following three possibilities.

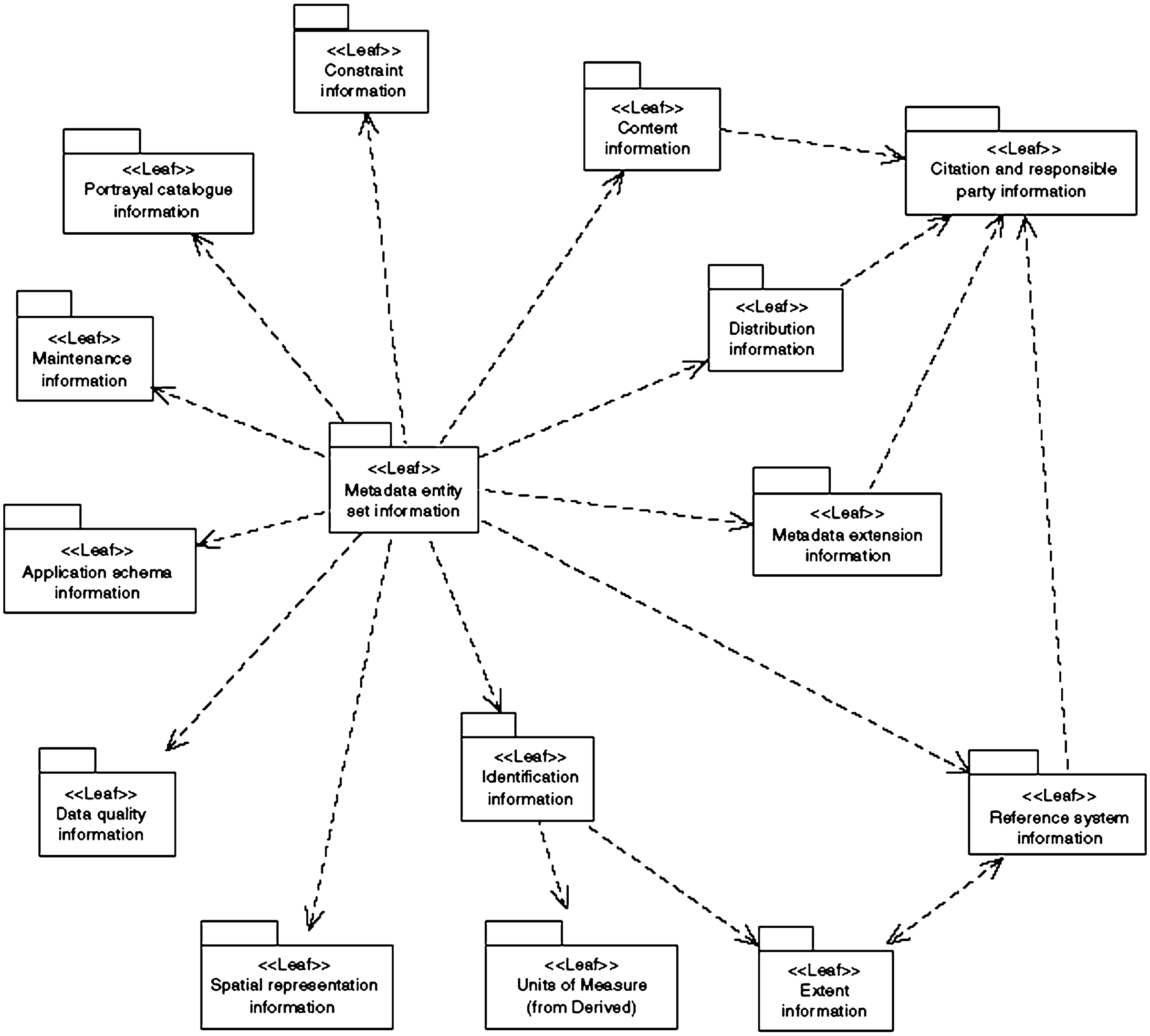
– Expressing a choice between two or more options. At least one option is mandatory and must be documented.
– Documenting a metadata entity or a metadata element if another element has been documented.
– Documenting a metadata element if a specific value for another metadata element has been documented. To facilitate reading by humans, plain text is used for the specific value. However, the code shall be used to verify the condition in an electronic user interface.

In short, if the answer to the condition is positive, then the metadata entity or the metadata element shall be mandatory.

Optional (O): The metadata entity or the metadata element may be documented or may not be documented. Optional metadata entities and optional metadata elements provide a guide to those looking to fully document their data. If an optional entity is not used, the elements contained within that entity (including mandatory elements) will also not be used. Optional entities may have mandatory elements; those elements only become mandatory if the optional entity is used.

ISO 19115 defines the core metadata that consists of a minimum set of metadata required to serve the full range of metadata applications. All the core elements must be available in a given metadata system. The optional ones need not be instantiated in a particular data set. These 22 metadata elements are shown in Table 1.

Currently, ISO is developing ISO 19115-2, which extends ISO 19115 for imagery and gridded data. Similar to the FGDC efforts and using FGDC CSDGM Extensions for Remote Sensing Metadata as its basis, ISO 19115-2 will define metadata elements particularly for imagery and gridded data within the framework of ISO 19115. According to the ISO TC 211 program of work, the final CD was posted in March 2007; barring major objection, it will be published as DIS in June 2007.



Catalogue Information Model, Figure 1 ISO 19115 Metadata UML Package

US NASA ECS Core Metadata Standard To enable an improved understanding of the Earth as an integrated system, in 1992 the National Aeronautics and Space Administration (NASA) of the United States of America started the Earth Observing System (EOS) program, which coordinates efforts to study the Earth as an integrated system. This program, using spacecraft, aircraft and ground instruments, allows humans to better understand climate and environmental changes and to distinguish between natural and human-induced changes. The EOS program includes a series of satellites, a science component, and a data system for long-term global observations of the land surface, biosphere, solid Earth, atmosphere, and oceans. The program aims at accumulating 15 years of Earth observation data at rate of over 2 terabytes per day. To support data archival, distribution, and management, NASA has developed an EOS Data and information System (EOSDIS) and its core system (ECS), the largest data and information system for Earth observation in the world.

In order to standardize the descriptions of data collected by the EOS program, NASA has developed the ECS Core Metadata Standard. The standard defines metadata in several areas: algorithm and processing packages, data sources, references, data collections, spatial and temporal extent, and content. The ECS Core Metadata Standard has been used as the catalogue information model for EOSDIS Data Gateway (EDG) and the EOS ClearingHOuse (ECHO). The ECS Core Metadata Standard was the basis

Catalogue Information Model, Table 1 ISO 19115 Core Metadata

Dataset title (M)	**Spatial representation type (O)**
(MD_Metadata > MD_DataIdentification.citation > CI_Citation.title)	(MD_Metadata > MD_DataIdentification.spatialRepresentationType)
Dataset reference date (M)	**Reference system (O)**
(MD_Metadata > MD_DataIdentification.citation > CI_Citation.date)	(MD_Metadata > MD_ReferenceSystem)
Dataset responsible party (O)	**Lineage (O)**
(MD_Metadata > MD_DataIdentification.pointOfContact > CI_ResponsibleParty)	(MD_Metadata > DQ_DataQuality.lineage > LI_Lineage)
Geographic location of the dataset (by four coordinates or by geographic identifier) (C)	**On-line resource (O)**
(MD_Metadata > MD_DataIdentification.extent > EX_Extent > EX_GeographicExtent > EX_GeographicBoundingBox or EX_GeographicDescription)	(MD_Metadata > MD_Distribution > MD_DigitalTransferOption.onLine > CI_OnlineResource)
Dataset language (M)	**Metadata file identifier (O)**
(MD_Metadata > MD_DataIdentification.language)	(MD_Metadata.fileIdentifier)
Dataset character set (C)	**Metadata standard name (O)**
(MD_Metadata > MD_DataIdentification.characterSet)	(MD_Metadata.metadataStandardName)
Dataset topic category (M)	**Metadata standard version (O)**
(MD_Metadata > MD_DataIdentification.topicCategory)	(MD_Metadata.metadata.StandardVersion)
Spatial resolution of the dataset (O)	**Metadata language (C)**
(MD_Metadata > MD_DataIdentification.spatial Resolution > MD_Resolution.equivalentScale or MD_Resolution.distance)	(MD_Metadata.language)
Abstact describing the dataset (M)	**Metadata character set (C)**
(MD_Metadata > MD_DataIdentification.abstact)	(MD_Metadata.characterSet)
Distribution format (O)	**Metadata point of contact (M)**
(MD_Metadata > MD_Distribution > MD_Format.name and MD_Format.version)	(MD_Metadata.contact > CI_ResponsibleParty)
Additional extent information for the dataset (vertical and temporal) (O)	**Metadata date stamp (M)**
(MD_Metadata > MD_DataIdentification.extent > EX_Extent > EX_TemporalExtent or EX_VerticalExtent)	(MD_Metadata.dateStamp)

for the development of FGDC CSDGM Extensions for Remote Sensing Metadata.

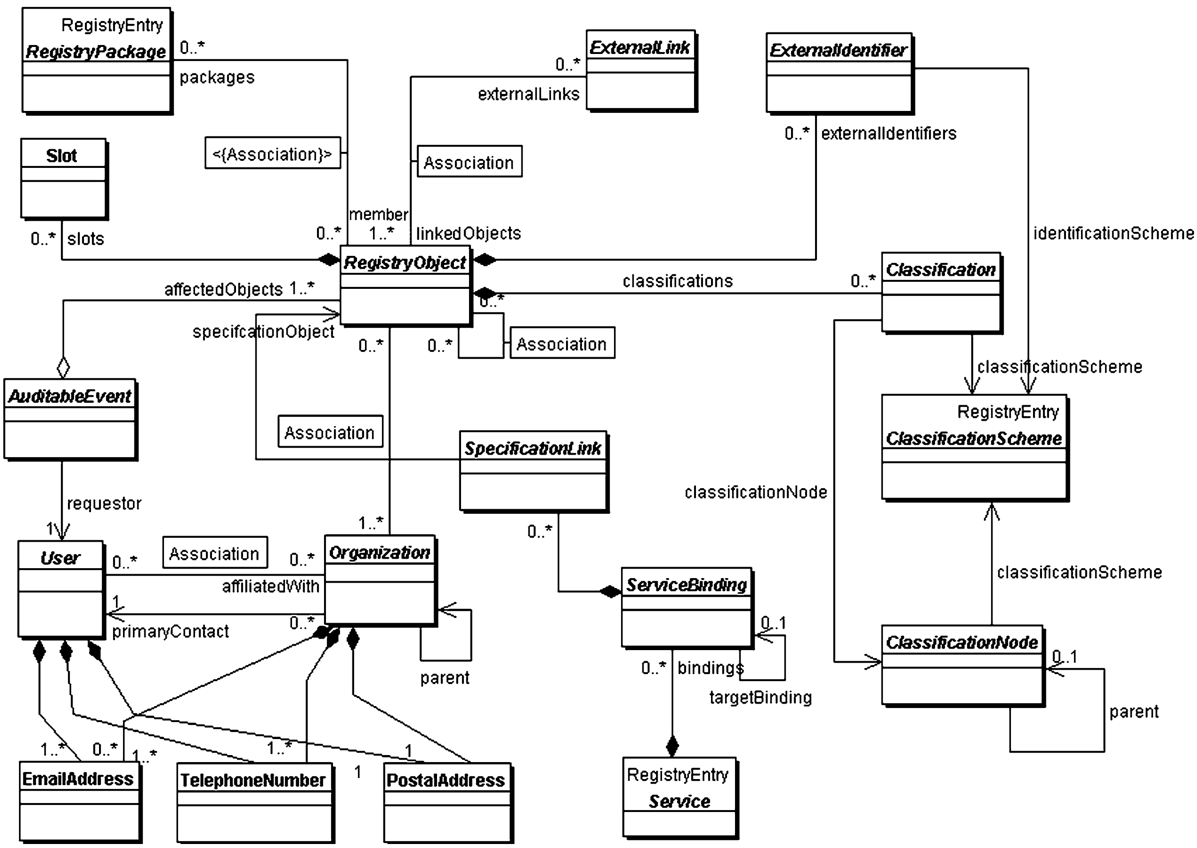
With new satellites being launched and instruments being operational, this standard will incorporate new keywords from them into new version. The current version is 6B, released on October 2002. The 6B version is logically segmented into eight modules for the purpose of readability, including Data Originator, ECS Collection, ECS Data Granule, Locality Spatial, Locality Temporal, Contact, Delivered Algorithm Package and Document [9].

ebRIM The ebXML Registry Information Model (ebRIM) was developed by OASIS to specify the information model for the ebXML *Registry* [2]. The goal of the ebRIM specification is to communicate what information is in the *Registry* and how that information is organized. The high level view of the information model is shown in Fig. 2.

ebRIM has been widely used in the world of e-business web services as the standardized information model for service registries. In the geospatial community, OGC has adopted the ebRIM specification as one of the application profiles of its Catalog Service for Web Specification [10]. And this model has been further approved as the preferred meta-model for future OGC CS-W Catalogue Application Profiles in the OGC technical meeting in December 2007, [11].

Future Directions

Catalogue information models define the organization of metadata in the catalogue. Each catalogue system normal-



Catalogue Information Model, Figure 2 The high level UML model of ebRIM [2]

ly adopts a metadata standard as its catalogue information model. With the wide acceptance of the concept of metadata, there are usually multiple related metadata standards used by different catalog systems in the same application domain. Metadata crosswalks are needed to build maps between two metadata elements and/or attributes so that the interoperability among the legacy catalogue systems becomes possible. Besides these cross-walks, another direction is to organize a formal representation of every metadata concept into a geospatial ontology for enabling semantic interoperability between catalogues.

In addition to the catalogue information models, there are other necessary parts to compose an operational catalogue service, such as query language, query and response protocol, etc. Related research has been directed in the OGC, FGDC, and other agencies.

Recommended Reading

1. OGC. OGC™ Catalogue Services Specification, OGC 04–021r3. http://portal.opengeospatial.org/files/?artifact_id=5929&Version=2. Accessed 13 Sept 2007
2. OASIS. ebXML Registry Information Model (ebRIM). http://www.oasis-open.org/committees/regrep/documents/2.0/specs/ebrim.pdf. Accessed 13 Sept 2007
3. Using Dublin Core. http://www.dublincore.org/documents/2001/04/12/usageguide/. Accessed 13 Sept 2007
4. Dublin Core. http://en.wikipedia.org/wiki/Dublin_Core. Accessed 13 Sept 2007
5. FGDC, CSDGM. http://www.fgdc.gov/standards/standards_publications/index_html. Accessed 13 Sept 2007
6. FGDC, Content Standard for Digital Geospatial Metadata, Extensions for Remote Sensing Metadata. http://www.fgdc.gov/standards/standards_publications/index_html. Accessed 13 Sept 2007
7. ISO/TC 211, ISO 19115 Geographic Information: Metadata
8. OGC. OpenGIS® Catalogue Services Specification 2.0 – ISO19115/ISO19119 Application Profile for CSW 2.0. OGC 04–038r2. https://portal.opengeospatial.org/files/?artifact_id=8305. Accessed 13 Sept 2007
9. ECS, Release 6B Implementation Earth Science Data Model for the ECS Project. http://spg.gsfc.nasa.gov/standards/heritage/eosdis-core-system-data-model. Accessed 13 Sept 2007
10. OGC. EO Products Extension Package for ebRIM (ISO/TS 15000–3) Profile of CSW 2.0. OGC 06–131. http://portal.opengeospatial.org/files/?artifact_id=17689. Accessed 13 Sept 2007

11. OGC. OGC Adopts ebRIM for Catalogues. http://www.opengeospatial.org/pressroom/pressreleases/655. Accessed 13 Sept 2007

Catalogue Information Schema

▶ Catalogue Information Model

Catalogue Metadata Schema

▶ Catalogue Information Model

Category, Geographic; RDF

▶ Geospatial Semantic Integration

Central Perspective

▶ Photogrammetric Methods

Central Projection

▶ Photogrammetric Methods

Centrographic Measures

▶ CrimeStat: A Spatial Statistical Program for the Analysis of Crime Incidents

CGI

▶ Web Mapping and Web Cartography

CGIS

▶ Geocollaboration

Chain

▶ Spatial Data Transfer Standard (SDTS)

Change Detection

JÉRÔME THÉAU
GIS Training and Research Center,
Idaho State University, Pocatello Idaho, ID, USA

C

Synonyms

Detection of changes; Digital change detection methods; Land cover change detection

Definition

Change Detection can be defined as the process of identifying differences in the state of an object or phenomenon by observing it at different times [1]. This process is usually applied to earth surface changes at two or more times. The primary source of data is geographic and is usually in digital format (e. g., satellite imagery), analog format (e. g., aerial photos), or vector format (e. g., feature maps). Ancillary data (e. g., historical, economic, etc.) can also be used.

Historical Background

Change detection history starts with the history of remote sensing and especially the first aerial photography taken in 1859 by Gaspard Felix Tournachon, also known as Nadar. Thereafter, the development of change detection is closely associated with military technology during world wars I and II and the strategic advantage provided by temporal information acquired by remote sensing. Civilian applications of change detection were developed following these events in the 20th century using mostly interpretation and analog means. However, civilian availability of data was limited until the 70's and 80's due to military classification of imagery.

The development of digital change detection era really started with the launch of Landsat-1 (called first: Earth Resources Technology Satellite) in July 1972. The regular acquisition of digital data of the earth surface in multispectral bands allowed scientists to get relatively consistent data over time and to characterize changes over relatively large area for the first time. The continuity of this mission as well as the launch of numerous other ones ensured the development of change detection techniques from that time.

However, the development of digital change detection techniques was limited by data processing technology capacities and followed closely the development of computer technologies. The situation evolves from the 1960's when a few places in the world were equipped with expensive computers to the present when personal computers are

fast and cheap enough to apply even complex algorithms and change detection techniques to satellite imagery. The computer technology also evolved from dedicated hardware to relatively user-friendly software specialized for image processing and change detection.
Based on published literature, the algebra techniques such as image differencing or image ratioing were the first techniques used to characterize changes in digital imagery during the 1970's [2]. These techniques are simple and fast to perform and are still widely used today. More complex techniques were developed since then with the improvement of processing capacities but also with the development of new theoretical approaches. Change detection analysis of the earth surface is a very active topic due to the concerns about consequences of global and local changes. This field of expertise is constantly progressing.

Scientific Fundamentals

Changes on Earth Surface

The earth surface is changing constantly in many ways. First, the time scales, at which changes can occur, are very heterogeneous. They may vary from catastrophic events (e. g., flood) to geological events (e. g., continental drift) which correspond to a gradient between punctual and continuous changes respectively. Secondly, the spatial scales, at which changes can occur, are also very heterogeneous and may vary from local events (e. g., road construction) to global changes (e. g., ocean water temperature). Due to this very large spatio-temporal range, the nature and extent of changes are complex to determine because they are interrelated and interdependent at different scales (spatial, temporal). Change detection is, therefore, a challenging task.

Imagery Characteristics Regarding Changes

Since the development of civilian remote sensing, the earth benefits from a continuous and increasing coverage by imagery such as: aerial photography or satellite imagery. This coverage is ensured by various sensors with various properties. First, in terms of the time scale, various *temporal resolutions* (i. e., revisit time) and mission continuities allow coverage of every point of the earth from days to decades. Secondly, in terms of the spatial scale, various spatial resolutions (i. e., pixel size, scene size) allow coverage of every point of the earth at a sub-meter to a kilometer resolution. Thirdly, sensors are designed to observe the earth surface using various parts of the electromagnetic spectrum (i. e., spectral domain) at different resolutions (i. e., spectral resolution). This diversity allows the characterization of a large spectrum of earth surface elements and change processes. However, change detection is still limited by data availability and data consistency (i. e., multi-source data).

Changes in Imagery

Changes in imagery between two dates translate into changes in radiance. Various factors can induce changes in radiance between two dates such as changes in: sensor calibration, solar angle, atmospheric conditions, seasons, or earth surface. The first premise of using imagery for change detection of the earth surface is that change in the earth surface must result in a change in radiance values. Secondly, the change in radiance due to earth surface changes must be large compared to the change in radiance due to other factors. A major challenge in change detection of the earth surface using imagery is to minimize these other factors. This is usually performed by carefully selecting relevant multidate imagery and by applying pre-processing treatments.

Data Selection and Pre-processing

Data selection is a critical step in change detection studies. The acquisition period (i. e., season, month) of multidate imagery is an important parameter to consider in image selection because it is directly related to phenology, climatic conditions, and solar angle. A careful selection of multidate images is therefore needed in order to minimize the effects of these factors. In vegetation change studies (i. e., over different years), for example, summer is usually used as the target period because of the relative stability of phenology, solar angle, and climatic conditions. The acquisition interval between multidate imagery is also important to consider. As mentioned before, earth surface changes must cause enough radiance changes to be detectable. However, the data selection is often limited by data availability and the choice is usually a compromise between the targeted period, interval of acquisition, and availability. The cost of imagery is also a limiting factor in data selection.
However, a careful data selection is usually not enough to minimize radiometric heterogeneity between multidate images. First, atmospheric conditions and solar angle differences usually need additional corrections and secondly other factors such as sensor calibration or geometric distortions need to be considered. In change detection analysis, multidate images are usually compared on a pixel basis. Then, very accurate *registrations* need to be performed between images in order to compare pixels at the same locations. Mis*registration* between multidate images can cause significant errors in change interpretation. The sensitivity of change detection approaches to mis*registration* is variable though. The minimization of radiometric hetero-

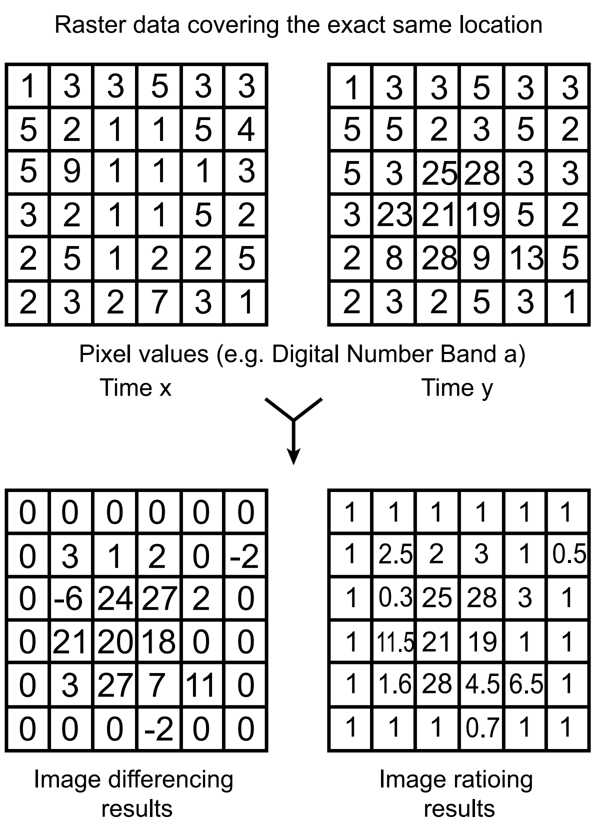
geneity (due to sources other than earth surface change) can be performed using different approaches depending on the level of correction required and the availability of atmospheric data. The techniques such as dark object subtraction, relative radiometric normalization or radiative transfer code can be used.

Change Detection Methods

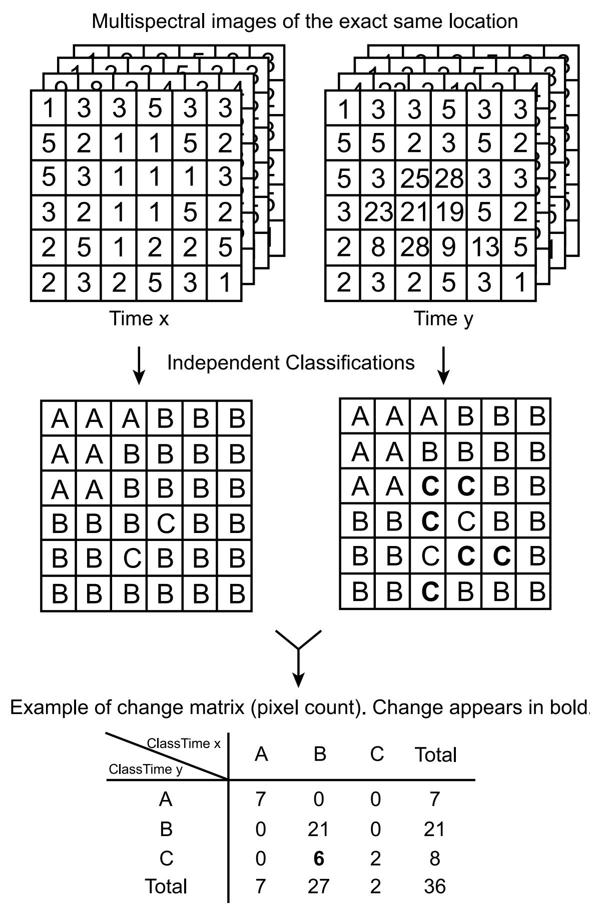
Summarized here are the most common methods used in change detection studies [1,2,3,4,5]. Most of these methods use image processing approaches applied to multidate satellite imagery.

Image differencing: This simple method is widely used and consists of subtracting registered images acquired at different times, pixel by pixel and band by band. No changes between times result in pixel values of 0, but if changes occurred these values should be positive or negative (Fig. 1). However, in practice, exact image *registration* and perfect radiometric corrections are never obtained for multidate images. Residual differences in radiance not caused by land cover changes are still present in images. Then the challenge of this technique is to identify threshold values of change and no-change in the resulting images. Standard deviation is often used as a reference values to select these thresholds. Different normalization, histogram matching, and standardization approaches are used on multidate images to reduce scale and scene dependent effects on differencing results. The image differencing method is usually applied to single bands but can be also applied to processed data such as multidate vegetation indices or principal components.

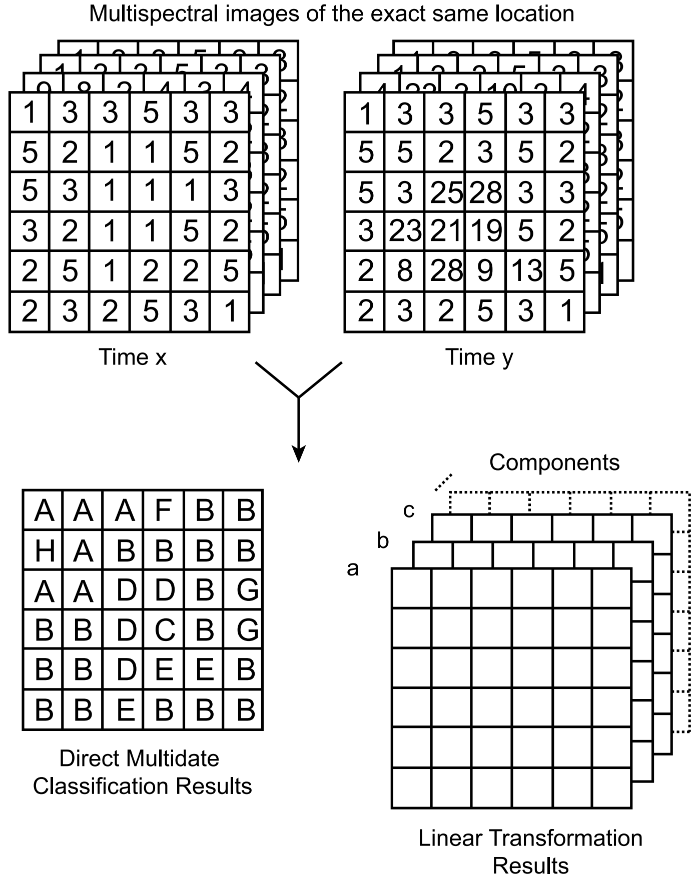
Image ratioing: This method is comparable to the image differencing method in terms of its simplicity and challenges. However, it is not as widely used. It is a ratio of registered images acquired at different times, pixel by pixel and band by band. Changes are represented by pixel values higher or lower than 1 (Fig. 1). Pixels with no change will have a value of one. In practice, for the same reasons as in image differencing, the challenge of this technique is in selecting threshold values between change and no change. This technique is often criticized because the non-normal



Change Detection, Figure 1 Example of image differencing and image ratioing procedures



Change Detection, Figure 2 Example of a post-classification procedure



Change Detection, Figure 3 Example of direct multidate classification and linear transformation procedures

distribution of results limits the validity of threshold selection using the standard deviation of resulting pixels.

Post-classification: This method is also commonly referred to as "Delta classification". It is widely used and easy to understand. Two images acquired at different times are independently classified and then compared. Ideally, similar thematic classes are produced for each classification. Changes between the two dates can be visualized using a change matrix indicating, for both dates, the number of pixels in each class (Fig. 2). This matrix allows one to interpret what changes occurred for a specific class. The main advantage of this method is the minimal impacts of radiometric and geometric differences between multidate images. However, the accuracy of the final result is the product of accuracies of the two independent classifications (e. g., 64% final accuracy for two 80% independent classification accuracies).

Direct multidate classification: This method is also referred to as "Composite analysis", "Spectral-temporal combined analysis", "Spectral-temporal change classification", "Multidate clustering", or "Spectral change pattern analysis". Multidate images are combined into a single dataset on which a classification is performed (Fig. 3). The areas of changes are expected to present different statistics (i. e., distinct classes) compared to the areas with no changes. The approach can be unsupervised or supervised and necessitates only one classification procedure. However, this method usually produces numerous classes corresponding to spectral changes within each single image but also to temporal changes between images. The interpretation of results is often complex and requires a good knowledge of the study area. Combined approaches using principal component analysis or Bayesian classifier can be performed to reduce data dimensionality or the coupling between spectral and temporal change respectively.

Linear transformations: This approach includes different techniques using the same theoretical basis. The Principal Component Analysis (PCA) and the Tasseled-Cap transformations are the most common ones. Linear transformations are often used to reduce spectral data dimensionality by creating fewer new components. The first components contain most of the variance in the data and are uncorrelated. When used for change detection purposes, linear transformations are performed on multidate images that are combined as a single dataset (Fig. 3).

After performing a PCA, unchanged areas are mapped in the first component (i. e., information common to multidate images) whereas areas of changes are mapped in the last components (i. e., information unique to either one of the different dates). Usually the PCA is calculated from a variance/co-variance matrix. However, standardized matrix (i. e., correlation matrix) is also used. The PCA is scene dependent and results can be hard to interpret. The challenging steps are to label changes from principal components and to select thresholds between change and no-change areas. A good knowledge of the study area is required.

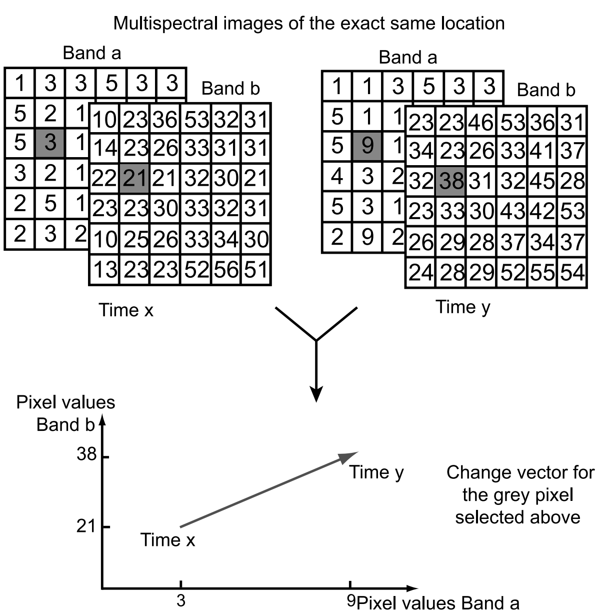
The Tasseled-Cap is also a linear transformation. However, unlike PCA, it is independent of the scene. The new component directions are selected according to pre-defined spectral properties of vegetation. Four new components are computed and oriented to enhance brightness, greenness, wetness, and yellowness. Results are also difficult to interpret and change labeling is challenging. Unlike PCA, Tasseled-Cap transformation for change detection requires accurate atmospheric calibration of multidate imagery.

Other transformations such as multivariate alteration detection or Gramm-Schmidt transformation were also developed but used to a lesser extent.

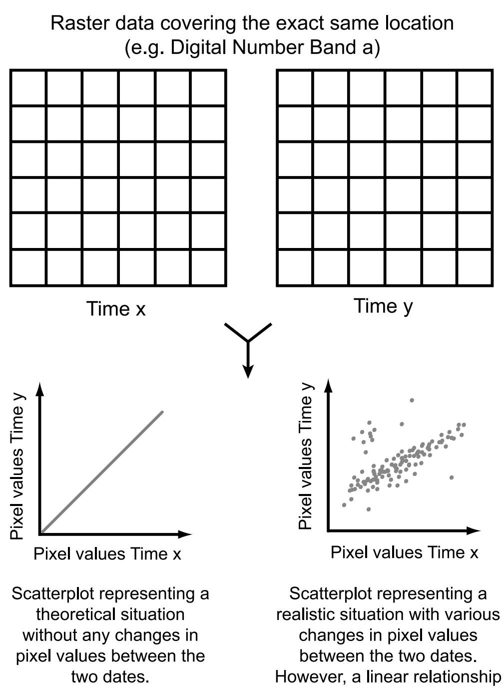
Change vector analysis: This approach is based on the spatial representation of change in a spectral space. When a pixel undergoes a change between two dates, its position in n-dimensional spectral space is expected to change. This change is represented by a vector (Fig. 4) which is defined by two factors, the direction which provides information about the nature of change and the magnitude which provides information about the level of change. This approach has the advantage to process concurrently any number of spectral bands. It also provides detailed information about change. The challenging steps are to define thresholds of magnitude, discriminating between change and no change, and to interpret vector direction in relation with the nature of change. This approach is often performed on transformed data using methods such as Tasseled-Cap.

Image regression: This approach assumes that there is a linear relationship between pixel values of the same area at two different times. This implies that a majority of the pixels did not encounter changes between the two dates (Fig. 5). A regression function that best describes the relationship between pixel values of each spectral band at two dates is developed. The residuals of the regression are considered to represent the areas of changes. This method has the advantage of reducing the impact of radiometric heterogeneity (i. e., atmosphere, sun angle, sensor calibration) between multidate images. However, the challenging steps are to select an appropriate regression function and to define thresholds between change and no change areas.

Multitemporal spectral mixture analysis: The spectral mixture analysis is based on the premise that a pixel reflectance value can be computed from individual values



Change Detection, Figure 4 Example and principle of the change vector procedure



Change Detection, Figure 5 Example and principle of the image regression procedure of its composing elements (i. e., end-members) weighted by their respective proportions. This case assumes a linear mixing of these components. This method allows retrieving sub-pixel information (i. e., surface proportions of end-members) and can be used for change detection purposes by performing separate analysis and comparing results at different dates (Fig. 6). The advantage of this method is to provide precise and repeatable results. The challenging step of this approach is to select suitable end-members.

Combined approaches: The previous techniques represent the most common approaches used for change detection purposes. They can be used individually, but are often combined together or with other image processing techniques to provide more accurate results. Numerous combinations can be used and they will not be described here. Some of them include the combination of vegetation indices and image differencing, change vector analysis and principal component analysis, direct multidate classification and principal component analysis, multitemporal spectral analysis and image differencing, or image enhancement and post-classification.

Example of change detection analysis: Mapping changes in caribou habitat using multitemporal spectral mixture analysis: The George River Caribou Herd (GRCH), located in northeastern Canada, increased from about 5,000 in the 1950s to about 700,000 head in the 1990s. This has led to an over-utilization of summer habitat, resulting in degradation of the vegetation cover. This degradation has had a direct impact on health problems observed in the caribou (*Rangifer tarandus*) population over the last few years and may also have contributed to the recent decline of the GRCH (404,000 head in 2000–2001). Lichen habitats are good indicators of caribou herd activity because of their sensitivity to overgrazing and over-trampling, their widespread distribution over northern territories, and their influence on herd nutrition. The herd range covers a very large territory which is not easily accessible. As a result, field studies over the whole territory are limited and aerial surveys cannot be conducted frequently. Satellite imagery offers the synoptic view and temporal resolution necessary for mapping and monitoring caribou habitat. In this example, a change detection approach using Landsat imagery was used. The procedure was based on spectral mixture analysis and produced maps showing the lichen proportion inside each pixel. The procedure was applied to multidate imagery to monitor the spatio-temporal evolution of the lichen resource over the past three decades and gave new information about the habitat used by the herd in the past, which was very useful to better understand population dynamics. Figure 6 summarizes the approach used in this study and illustrates the steps typical of a change detection procedure.

Key Applications

The earth surface is changing constantly in many ways. Changes occur at various spatial and temporal scales in numerous environments. Change detection techniques are employed for different purposes such as research, management, or business [2,6,7,8,9,10,11,12]. Monitoring changes using GIS and remote sensing is therefore used in a wide field of applications. A non-exhaustive list of key applications is presented here.

Forestry

- Deforestation (e. g., clear cut mapping, regeneration assessment)
- Fire monitoring (e. g., delineation, severity, detection, regeneration)
- Logging planning (e. g., infrastructures, inventory, biomass)
- Herbivory (e. g., insect defoliation, grazing)
- Habitat fragmentation (e. g., landcover changes, heterogeneity)

Agriculture and Rangelands

- Crop monitoring (e. g., growing, biomass)
- Invasive species (e. g., detection, distribution)
- Soil moisture condition (e. g., drought, flood, landslides)
- Desertification assessment (e. g., bare ground exposure, wind erosion)

Urban

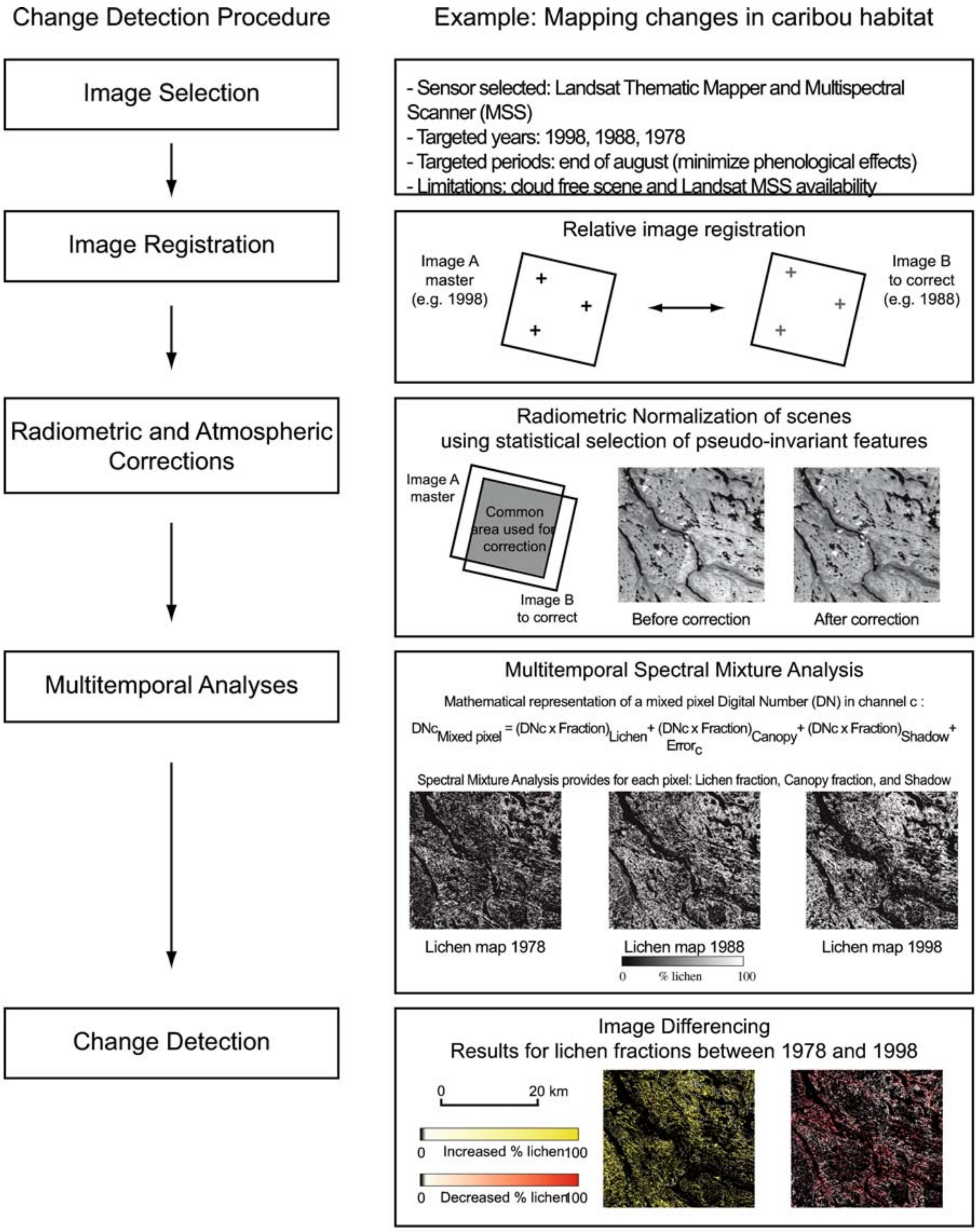
- Urban sprawl (e. g., urban mapping)
- Transportation and infrastructure planning (e. g., landcover use)

Ice and Snow

- Navigation route (e. g., sea ice motion)
- Infrastructure protection (e. g., flooding monitoring)
- Glacier and ice sheet monitoring (e. g., motion, melting)
- Permafrost monitoring (e. g., surface temperature, tree line)

Ocean and Coastal

- Water quality (e. g., temperature, productivity)
- Aquaculture (e. g., productivity)
- Intertidal zone monitoring (e. g., erosion, vegetation mapping)
- Oil spill (e. g., detection, oil movement)



For more details see: Théau and Duguay (2004) Mapping Lichen Habitat Changes inside the Summer Range of the George River Caribou Herd (Québec-Labrador, Canada) using Landsat Imagery (1976-1998). Rangifer. 24: 31-50.

Change Detection, Figure 6 Example of a change detection procedure. Case study of mapping changes in caribou habitat using multitemporal spectral mixture analysis

Future Directions

In the past decades, a constant increase of remotely sensed data availability was observed. The launch of numerous satellite sensors as well as the reduction of product costs can explain this trend. The same evolution is expected in the future. The access to constantly growing archive contents also represents a potential for the development of more change detection studies in the future. Long-term missions such as Landsat, SPOT (Satellite pour l'Observation de la Terre), AVHRR (Advanced Very High Resolution Radiometer) provide continuous data for more than 20 to 30 years now. Although radiometric heterogeneity between sensors represents serious limitation in time series analysis, these data are still very useful for long term change studies. These data are particularly suitable in the development of temporal trajectory analysis which usually involves the temporal study of indicators (e. g., vegetation indices, surface temperature) on a global scale.

Moreover, as mentioned before in the Historical Background section, the development of change detection techniques are closely linked with the development of computer technologies and data processing capacities. In the future, these fields will still evolve in parallel and new developments in change detection are expected with the development of computer technologies.

Developments and applications of new image processing methods and geospatial analysis are also expected in the next decades. Artificial intelligence systems as well as knowledge-based expert systems and machine learning algorithms represent new alternatives in change detection studies [3]. These techniques have gained considerable attention in the past few years and are expected to increase in change detection approaches in the future. One of the main advantages of these techniques is that they allow the integration of existing knowledge and non-spectral information of the scene content (e. g., socio-economic data, shape, and size data). With the increasing interest in using integrated approaches such as coupled human-environment systems, these developments look promising. The recent integration of change detection and spatial analysis modules in most GIS software also represents a big step towards integrated tools in the study of changes on the earth surface. This integration also includes an improvement of compatibility between image processing software and GIS software. More developments are expected in the future which will provide new tools for integrating multi-source data more easily (e. g., digital imagery, hard maps, historical information, vector data).

Cross References

► Co-location Pattern Discovery
► Correlation Queries in Spatial Time Series Data

Recommended Reading

1. Singh, A.: Digital change detection techniques using remotely-sensed data. Int. J. Remote Sens. **10**(6), 989–1003 (1989)
2. Lunetta, R.S., Elvidge, C.D.: Remote sensing change detection: environmental monitoring methods and applications. Ann Arbor Press, Chelsea, USA, (1998) p. 318
3. Coppin, P., Jonckheere, I., Nackaerts, K., Muys, B., Lambin, E.: Digital change detection methods in ecosystem monitoring: a review. Int. J. Remote Sens. **25**(9), 1565–1596 (2004)
4. Lu, D., Mausel, P., Brondízios, E., Moran, E.: Change detection techniques. Int. J. Remote Sens. **25**(12), 2365–2407 (2004)
5. Mas, J.-F.: Monitoring land-cover changes: a comparison of change detection techniques. Int. J. Remote Sens. **20**(1), 139–152 (1999)
6. Canada Centre for Remote Sensing. http://ccrs.nrcan.gc.ca/index_e.php. Accessed November 2006
7. Diversitas – Integrating biodiversity science for human well-being. http://www.diversitas-international.org/. Accessed November 2006
8. ESA – Observing the Earth. http://www.esa.int/esaEO/index.html. Accessed November 2006
9. Global Change Master Directory. http://gcmd.nasa.gov/index.html. Accessed November 2006
10. IGBP – International Geosphere-Biosphere Programme. http://www.igbp.net/. Accessed November 2006
11. IHDP – International Human Dimensions Programme on Global Environmental Change. http://www.ihdp.org/. Accessed November 2006
12. WCRP – World Climate Research Programme. http://wcrp.wmo.int/. Accessed November 2006

Change of Support Problem

► Error Propagation in Spatial Prediction

Channel Modeling and Algorithms for Indoor Positioning

Muzaffer Kanaan, Bardia Alavi,
Ahmad Hatami, Kaveh Pahlavan
Center for Wireless Information Network Studies,
Worcester Polytechnic Institute, Worcester, MA, USA

Synonyms

Indoor geolocation; Indoor location estimation; Indoor position estimation

Definition

One of the new frontiers in wireless networking research is location awareness. Knowledge of a user's location enables a number of location-based services (LBS) to be delivered to that user. However, while this problem has been largely addressed for the outdoor environment, indoor positioning is an open area of research. Here, the problem of accurate indoor positioning is discussed, and the current state of the art in accurate position estimation techniques is reviewed.

Historical Background

Serious research in the field of positioning first began in the 1960s, when several US government agencies, including the Department of Defense (DoD), National Aeronautics and Space Administration (NASA) and the Department of Transportation (DOT) expressed interest in developing systems for position determination [1]. The result, known as the Global Positioning System (GPS), is the most popular positioning system in use today. Activity in this area continued after cellular networks flourished in the 1990s, driven largely by regulatory requirements for position estimation, such as E-911.

While these developments were taking place, similar research and development activity started in the field of indoor positioning, due to emerging applications in the commercial as well as public safety/military areas. In the commercial space, indoor positioning is needed for applications such as tracking people with special needs (such as people who are sight-impaired), as well as locating equipment in warehouses and hospitals. In the public safety and military space, very accurate indoor positioning is required to help emergency workers as well as military personnel effectively complete their missions inside buildings. Some of these applications also require simple, low-power user terminals such as those that might be found in ad-hoc sensor networks.

Positioning techniques developed for GPS and cellular networks generally do not work well in indoor areas, owing to the large amount of signal attenuation caused by building walls. In addition, the behavior of the indoor radio channel is very different from the outdoor case, in that it exhibits much stronger multipath characteristics. Therefore, new methods of position estimation need to be developed for the indoor setting. In addition, the accuracy requirements of indoor positioning systems are typically a lot higher. For an application such as E-911, an accuracy of 125 m for 67% of the time is considered acceptable [2], while a similar indoor application typically requires an accuracy level on the order of only a few meters [3]. In the next few sections, an overview of positioning techniques is provided for the indoor environment.

Scientific Fundamentals

Structure of a Positioning System

The basic structure of a positioning system is illustrated in Fig. 1 below, where a sensor, (whose location is to be determined), is shown. The system consists of two parts: reference points (RPs) and the positioning algorithm. The RPs are radio transceivers, whose locations are assumed to be known with respect to some coordinate system. Each RP measures various characteristics of the signal received from the sensor, which is referred to in this entry as *location metrics*. These location metrics are then fed into the positioning algorithm, which then produces an estimate of the location of the sensor.

The location metrics are of three main types:

- Angle of Arrival (AOA)
- Time of arrival (TOA)
- Received signal strength (RSS)

This section is organized in four subsections; in the first three, each of these location metrics is discussed in greater detail, while the last is devoted to a nonexhaustive survey of position estimation techniques using these metrics.

Angle of Arrival

As its name implies, AOA gives an indication of the direction the received signal is coming from. In order to estimate the AOA, the RPs need to be equipped with special antennae arrays. Figure 2 shows an example of the AOA estimation in an ideal nonmultipath environment. The two RPs measure the AOAs from the sensor as 78.3° and 45° respectively. These measurements are then used to form lines of position, the intersection of which is the position estimate.

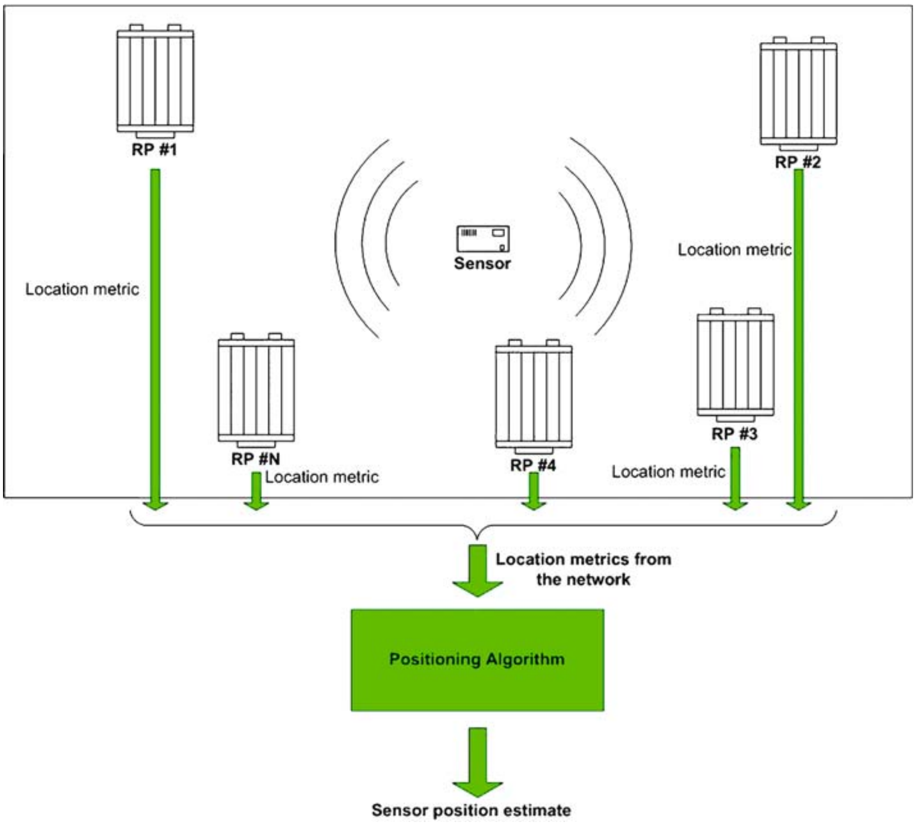
In real-world indoor environments, however, multipath effects will generally result in AOA estimation error. This error can be expressed as:

$$\hat{\theta} = \theta_{\text{true}} \mp \alpha \tag{1}$$

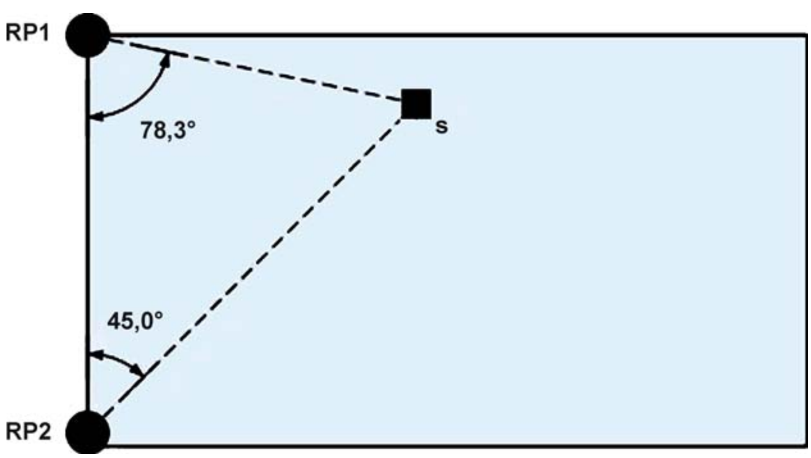
where θ_{true} is the true AOA value, generally obtained when the sensor is in the line-of-sight (LOS) path from the RP. In addition, $\hat{\theta}$ represents the estimated AOA, and α is the AOA estimation error. As a result of this error, the sensor position is restricted over an area defined with an angular spread of 2α, as illustrated in Fig. 3 below for the two-RP scenario. This clearly illustrates that in order to use AOA for indoor positioning, the sensor has to be in the LOS path to the RP, which is generally not possible.

Time of Arrival (TOA)

TOA gives an indication of the range (i. e., distance between a transmitter and a receiver). The basic concept can be illustrated with reference to the channel profile of Fig. 4 below. Since the speed of light in free space, c, is constant, the TOA of the direct path (DP) between the transmitter and the receiver, τ, will give the true range



Channel Modeling and Algorithms for Indoor Positioning, Figure 1 General structure of an indoor geolocation system. *RP* Reference point



Channel Modeling and Algorithms for Indoor Positioning, Figure 2 Illustration of angle of arrival (AOA). *S* Sensor

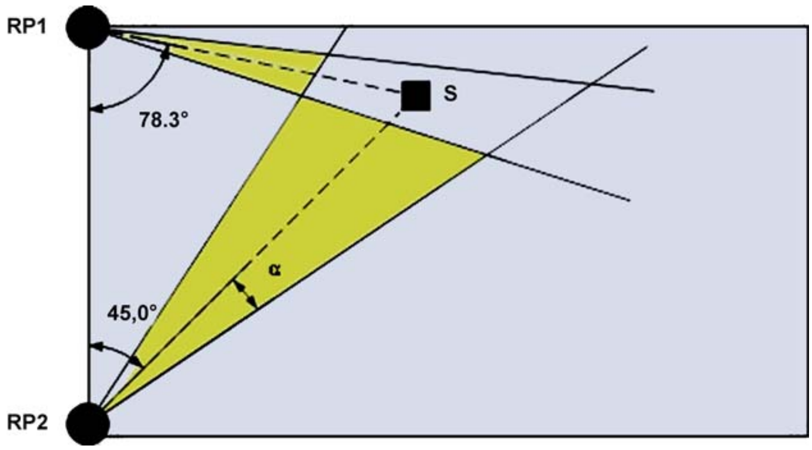
between the transmitter and receiver as defined by the equation:

$$d = c \times \tau\,. \quad (2)$$

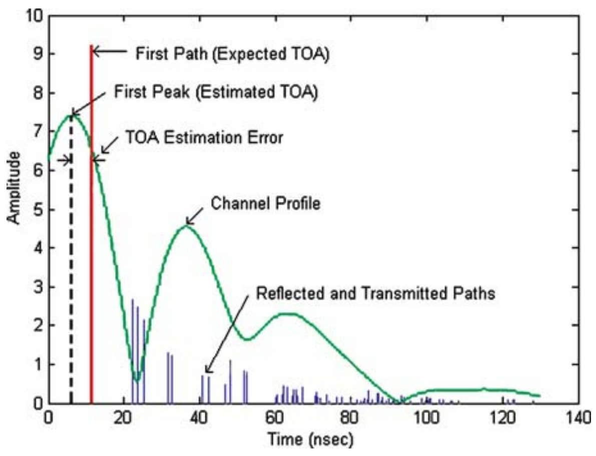
In practice, the TOA of the DP cannot be estimated perfectly, as illustrated in Fig. 4. The result is *ranging error* [also referred to as the *distance measurement error* (DME) in the literature], given as:

$$\varepsilon = \hat{d} - d \quad (3)$$

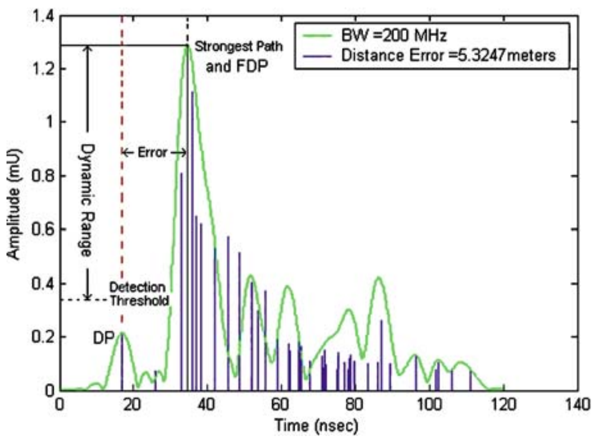
where $\hat{d}$ is the estimated distance and d is the true distance. There are two main sources of ranging error: multipath effects and undetected direct path (UDP) conditions. Multipath effects will result in the DP, as well as reflected and transmitted paths to be received. It has been shown empirically that multipath ranging error can be reduced by increasing the bandwidth of the system used for the TOA estimation [4]. UDP conditions, on the other hand, refer to cases where the DP cannot be detected at all, as shown in



Channel Modeling and Algorithms for Indoor Positioning, Figure 3 Illustration of AOA in the presence of multipath



Channel Modeling and Algorithms for Indoor Positioning, Figure 4 Illustrating basic time of arrival (TOA) principles for positioning



Channel Modeling and Algorithms for Indoor Positioning, Figure 5 Illustration of undetected direct path (UDP)-based distance measurement error (DME) at a bandwith (BW) of 200 MHz. *FDP* stands for first detected path

Fig. 5 below. UDP conditions generally occur at the edge of coverage areas, or in cases where there are large metallic objects in the path between the transmitter and the receiver. As a result, the difference between the first detected path (FDP) and the DP is beyond the dynamic range of the receiver, and the DP cannot be detected, as shown in Fig. 5. Unlike multipath-based ranging error, UDP-based ranging error typically cannot be reduced by increasing the bandwidth. In addition, the occurrence of UDP-based ranging error is itself random in nature [4].

Through UWB measurements in typical indoor areas, it has been shown that both multipath ranging error and UDP-based ranging error follow a Gaussian distribution, with mean and variance that depends on the bandwidth of operation [4]. The overall model can be expressed as follows:

$$\hat{d} = d + G(m_{\mathrm{w}}, \sigma_{\mathrm{w}}) \log(1+d) + \zeta \cdot G(m_{\mathrm{UDP,w}}, \sigma_{\mathrm{UDP,w}}) \quad (4)$$

where $G(m_{\mathrm{w}}, \sigma_{\mathrm{w}})$ and $G(m_{\mathrm{UDP,w}}, \sigma_{\mathrm{UDP,w}})$ are the Gaussian random variable (RV) that refer to multipath and UDP-based ranging error, respectively. The subscript w in both cases denotes the bandwidth dependence. The parameter ζ is a binary RV that denotes the presence or absence of UDP conditions, with a probability density function (PDF) given as:

$$f(\zeta) = \left(1 - P_{\mathrm{UDP,w}}\right) \delta(\zeta - 1) + P_{\mathrm{UDP,w}} \delta(\zeta) \quad (5)$$

where $P_{\mathrm{UDP,w}}$ denotes the probability of occurrence of UDP-based ranging error.

Received Signal Strength

RSS is a simple metric that can be measured and reported by most wireless devices. For example, the MAC layer of IEEE 802.11 WLAN standard provides RSS information from all active access points (APs) in a quasiperiodic beacon signal that can be used as a metric for positioning [5]. RSS can be used in two ways for positioning purposes.

If the RSS decays linearly with the log-distance between the transmitter and receiver, it is possible to map an observed RSS value to a distance from a transmitter and consequently determine the user's location by using distances from three or more APs. In other words:

$$RSS_{\mathrm{d}} = 10\log_{10} P_{\mathrm{r}} = 10\log_{10} P_{\mathrm{t}} - 10\alpha \log_{10} d + X \quad (6)$$

where α is the distance–power gradient, X is the shadow fading (a lognormal distributed random variable), P_{r} is the received power, and P_{t} is the transmitted power. While simple, this method yields a highly inaccurate estimate of distance in indoor areas, since instantaneous RSS inside a building varies over time, even at a fixed location; this is largely due to shadow fading and multipath fading. If, on the other hand, the RSS value to expect at a given point in an indoor area is known, then the location can be estimated as the point where the expected RSS values approximate the observed RSS values most closely. This is the essence of the *pattern recognition* approach to position estimation, which will be discussed in greater detail in the following section.

Position Estimation Techniques

Position estimation techniques can be categorized in a number of different ways. They can be grouped in terms of whether the sensing infrastructure used for measuring location metrics is deployed in a fixed or an ad-hoc manner. They can also be grouped according to how the position computations are performed. In the category of *centralized algorithms*, all the location metrics are sent to one central node, which then carries out the computations. In contrast, the term *distributed algorithms* refers to a class of algorithms where the computational load for the position calculations are spread out over all the nodes in the network. In the next few sections, some examples of centralized and distributed positioning algorithms for both fixed positioning as well as ad-hoc scenarios will be discussed. Owing to space limitations, the treatment is by no means exhaustive; the interested reader is referred to [6] and [7] as well as any associated references contained therein.

Centralized Algorithms

In this section, two algorithms for fixed position estimation, and one algorithm from ad-hoc positioning is discussed. For fixed location estimation, the closest neighbor with TOA Grid (CN-TOAG) [8], as well as ray-tracing assisted closest neighbor (RT-CN) algorithms [9] are discussed. For ad-hoc positioning, a distributed version of the least-squares (LS) algorithm is presented [10].

CN-TOAG Algorithm

The CN-TOAG algorithm leverages the fact that at any given point in an indoor covered by a number of RPs, the exact value of the TOA is known [8]. Consider the grid arrangement of RPs in an indoor setting, as shown in Fig. 6. Each of these RPs would perform a range measurement, $d_i(1 < qi < qN$, where N is the number of RPs in the grid) to the user to be located.

Let $\mathbf{D}$ represent the vector of range measurements that are reported by the RPs, and let $\mathbf{Z}$ represent the vector of expected TOA-based range measurements at a certain point, $\mathbf{r} = (x, y)$. For the purposes of this algorithm, $\mathbf{Z}$ is known as the *range signature* associated with the point $\mathbf{r}$. An estimate of the user's location, $\hat{\mathbf{r}}$, can be obtained by finding that point $\mathbf{r}$, where $\mathbf{Z}$ most closely approximates $\mathbf{D}$. The error function, $\mathrm{e}(\mathbf{r}) = \mathrm{e}(x, y)$, is defined as:

$$e(\mathbf{r}) = e(x, y) = ||\mathbf{D} - \mathbf{Z}(\mathbf{r})|| = ||\mathbf{D} - \mathbf{Z}(x, y)|| \quad (7)$$

where $||.||$ represents the vector norm. Equation (7) can also be written as:

$$e(x, y) = \sqrt{\sum_{k=1}^{N} \left(d_k - \sqrt{(x - X_k)^2 + (y - Y_k)^2} \right)^2} \quad (8)$$

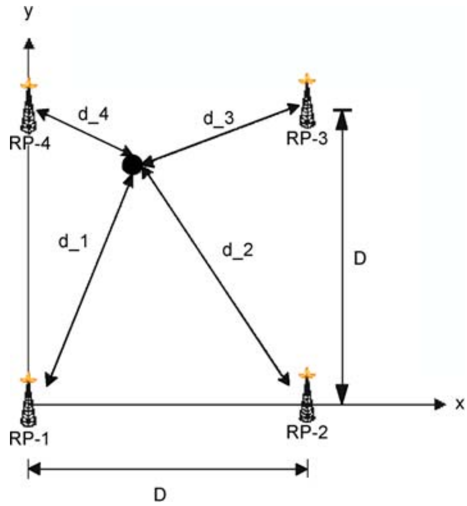
where N is the number of RPs, d_k is the range measurement performed by kth RP $(1 < qk < qN)$, and (X_k, Y_k) represents the location of the kth RP in Cartesian coordinates (assumed to be known precisely). The estimated location of the mobile, $\hat{\mathbf{r}}$, can then be obtained by finding the point (x, y) that minimizes (8). This point can be found by using the gradient relation:

$$\nabla e(x, y) = \mathbf{0} \quad (9)$$

Owing to the complexity of the function in (2), it is not possible to find an analytical solution to this problem. CN-TOAG provides a numerical method of solving (9), detailed in [8].

RT-CN Algorithm

The RT-CN algorithm is based on the RSS metric. The general idea is that the RSS characteristics of the area covered by the RPs are characterized in a data structure known as a *radio map*. Generally, the radio map is generated using on-site measurement in a process called training or fingerprinting. On-site measurement is a time- and labor-consuming process in a large and dynamic indoor environment. In [9] two alternative methods to generate a radio map without on-site measurements are introduced. The RT-CN algorithm uses two-dimensional ray-tracing (RT) computations to generate the reference radio map. During localization mobile station (MS) applies the nearest neighbor (NN) algorithm to the simulated radio map and the point that is the closest in signal space to the observed RSS values. In this way, a very high resolution radio map can be generated and higher localization accuracy results. In order to generate an accurate radio map in this technique, the localization system requires knowledge of the location of access points within the coverage area. In addition, a powerful central entity is required, both to perform the RT computations for the radio map, and to execute the NN algorithm on the radio map to come up with the final position estimate. As such, it is an example of a centralized pattern recognition algorithm.



Channel Modeling and Algorithms for Indoor Positioning, Figure 6
System scenario for performance evaluation

Distributed LS Algorithm

The algorithm that is featured in [10] is a distributed implementation of the steepest descent LS algorithm. The system scenario assumes ultrawide band (UWB) communications between sensor nodes. The sensor nodes perform range measurements between themselves and all the neighbors that they are able to contact. Then the following objective function is minimized using the distributed LS algorithm:

$$E = \frac{1}{2} \sum_{i} \sum_{j \in N(i)} \left(d_{ij} - \hat{d}_{ij} \right)^2 \tag{10}$$

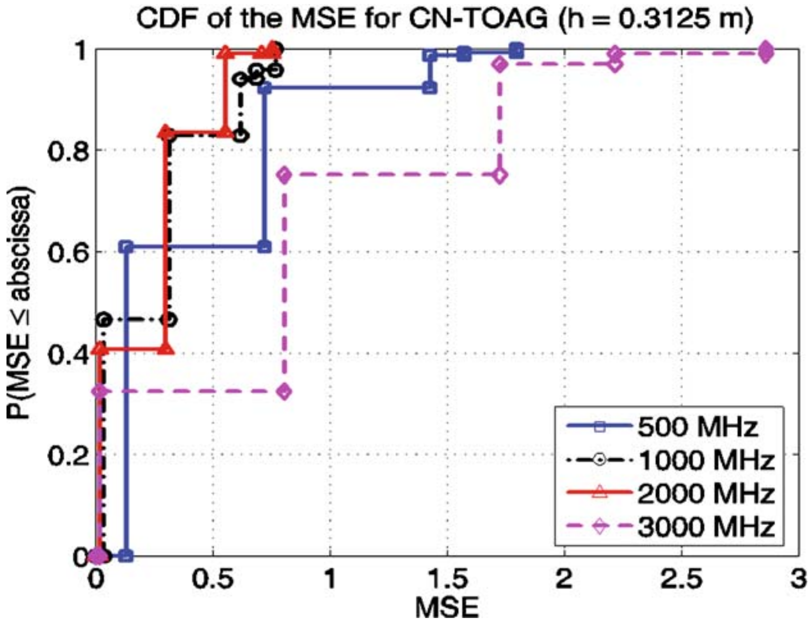
where d_{ij} is the actual distance between two nodes i and j and $\hat{d}_{ij}$ is the estimated distance between the same two nodes. Assuming some transmission range R for every sensor node, $N(i)$ represents the set of neighbors for node i, i. e., $N(i) = \left\{ j : d_{ij} < qR, i \neq j \right\}$.

Effects of the Channel Behavior on TOA-Based Positioning Algorithm Performance

Channel behavior is intimately linked with the performance of the positioning algorithms. As already noted above, the main effect of the channel is to introduce errors into the measurement of the metrics used for the positioning process. The precise manner in which these errors are introduced is determined by the quality of link (QoL) between the sensor and all the RPs that it is in contact with. In a TOA-based system, the exact amount of error from a given RP depends on whether UDP conditions exist or not. In this case, the channel is said to exhibit *bipolar* behavior, i. e., it suddenly switches from the detected direct path (DDP) state to the UDP state from time to time and this results in large DME values. These will then translate to large values of estimation error; in other words, the quality of estimation (QoE) will be degraded [11].

Owing to the site-specific nature of indoor radio propagation, the very occurrence of UDP conditions is random and is best described statistically [4]. That being the case, the QoE (i. e., location estimation accuracy) will also need to be characterized in the same manner. Different location-based applications will have different requirements for QoE. In a military or public-safety application (such as keeping track of the locations of fire-fighters or soldiers inside a building), high QoE is desired. In contrast, lower QoE might be acceptable for a commercial application (such as inventory control in a warehouse). In such cases, it is essential to be able to answer questions like: "What is the probability of being able to obtain a mean square error (MSE) of 1 m^2 from an algorithm x over different building environments that give rise to different amounts of UDP?" or "What algorithm should be used to obtain an MSE of 0.1 cm^2 over different building environments?" Answers to such questions will heavily influence the design, operation and performance of indoor geolocation systems.

Given the variability of the indoor propagation conditions, it is possible that the distance measurements performed by some of the RPs will be subject to DDP errors, while some will be subject to UDP-based errors. Various combinations of DDP and UDP errors can be observed. To illustrate, consider the example system scenario shown in Fig. 6. For example, the distance measurements performed by RP-1 may be subject to UDP-based DME, while the measure-



Channel Modeling and Algorithms for Indoor Positioning, Figure 7 (MSE) Profile for the closest- neighbor with TOA grid (CN-TOAG) algorithm

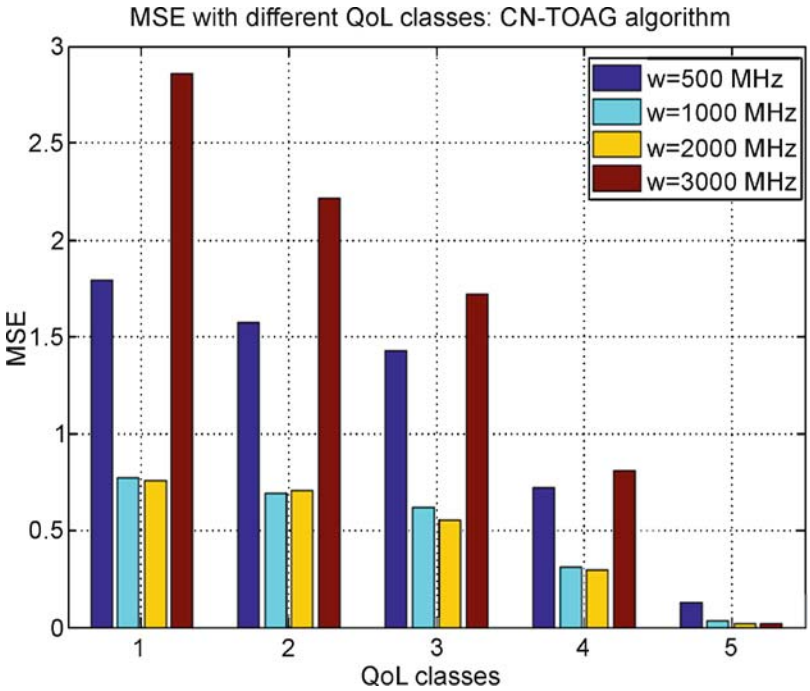
ments performed by the other RPs may be subject to DDP-based DME; this combination can be denoted as *UDDD*. Other combinations can be considered in a similar manner. Since the occurrence of UDP conditions is random, the performance metric used for the location estimate (such as the MSE) will also vary stochastically and depends on the particular combination observed. For the four-RP case shown in Fig. 6, it is clear that the following distinct combinations will have to be used: *UUUU*, *UUUD*, *UUDD*, *UDDD*, and *DDDD*. Each of these combinations can be used to characterize a different *QoL class*. The occurrence of each of these combinations will give rise to a certain MSE value in the location estimate. This MSE value will also depend on the specific algorithm used. There may be more than one way to obtain each DDP/UDP combination. If UDP conditions occur with probability P_{udp}, then the overall probability of occurrence of the ith combination P_{i} can be generally expressed as:

$$P_i = \begin{pmatrix} N \\ N_{\text{udp,i}} \end{pmatrix} P_{\text{udp}}^{N_{\text{udp,i}}} \left(1 - P_{\text{udp}}\right)^{N - N_{\text{udp,i}}} \qquad (11)$$

where N is the total number of RPs (in this case four), and $N_{\text{udp,i}}$ is the number of RPs where UDP-based DME is observed. Combining the probabilities, P_{i} with the associated MSE values for each QoL class, a discrete cumulative distribution function (CDF) of the MSE can be obtained. This discrete CDF is known as the *MSE profile* [11]. The use of the MSE profile will now be illustrated with examples, focusing on the CN-TOAG algorithm.

The system scenario in Fig. 6 is considered with $D = 20$ m. A total of 1,000 uniformly distributed random sensor locations are simulated for different bandwidth values. In line with the FCC's formal definition of UWB signal bandwidth as being equal to or more than 500 MHz [12], the results are presented for bandwidths of 500, 1,000, 2,000, and 3,000 MHz. For each bandwidth value, different QoL classes are simulated, specifically *UUUU*, *UUUD*, *UUDD*, *UDDD*, *DDDD*. Once a sensor is randomly placed in the simulation area, each RP calculates TOA-based distances to it. The calculated distances are then corrupted with UDP- and DDP-based DMEs in accordance with the DME model based on UWB measurements as given in [4]. The positioning algorithm is then applied to estimate the sensor location. Based on 1,000 random trials, the MSE is calculated for each bandwidth value and the corresponding combinations of UDP- and DDP-based DMEs. The probability of each combination is also calculated in accordance with (11).

The results are shown in Figs. 7 and 8. Figure 7 shows the MSE profiles for the CN-TOAG algorithm. From this plot, it is observed that as the bandwidth increases from 500 MHz to 2,000 MHz, the range of MSE profile values gets smaller. This correlates with the findings of [4], where it was observed that the overall DME goes down over this specific range of bandwidths. Above 2,000 MHz, however, the MSE profile becomes wider as a result of increased probability of UDP conditions [4], which increases the overall DME. This, in turn, translates into an increase in the position estimation error. In order to gain further insight into the variation of the QoE across the different



Channel Modeling and Algorithms for Indoor Positioning, Figure 8 Quality of link (QoE) variation across the various QoL classes

QoL classes, again considering bandwidth as a parameter, just the MSE is plotted, as seen in Fig. 8.

Key Applications

The applications of indoor localization technology are vast, and can be broadly classified into two categories: commercial and public safety/military. Commercial applications range from inventory tracking in a warehouse to tracking children, the elderly and people with special needs [13]. Location-sensitive web-browsing, and interactive tour guides for museums are other examples [14]. In the public safety/military space, the most prevalent application is to help emergency workers (police, firefighters, etc.).

Accurate indoor localization is also an important part of various personal robotics applications [15] as well as in the more general context of context-aware computing [16]. More recently, location sensing has found applications in location-based handoffs in wireless networks [17], location-based ad-hoc network routing [18] and location-based authentication and security. Many of these applications require low-cost, low-power terminals that can be easily deployed with little or no advanced planning; this is the basis for developments in ad-hoc sensor networks. Recent developments in integrated circuit (IC) technology as well as microelectromechanical systems (MEMS) have made it possible to realize such low-cost, low-power terminals. In the next few years, there will undoubtedly be scores of new applications for indoor localization.

Future Directions

Indoor positioning is a relatively new area of research and, as such, there are a number of different problems to be solved. Among these are questions such as: "What algorithms and techniques should be used to obtain a certain level of positioning error performance?" and "What is the best performance that can be obtained from a given positioning algorithm under UDP conditions?". Issues such as these will need to be looked at in order for the indoor positioning field to mature.

Cross References

► Indoor Positioning

Recommended Reading

1. Kaplan, E.D.: Understanding GPS. Artech, Boston (1996)
2. FCC Docket No. 94–102. Revision of the commissions rules to insure compatibility with enhanced 911 emergency calling systems. Federal Communications Commission Tech. Rep. RM-8143, July 1996
3. Sayed, A.H., Tarighat, A., Khajehnouri, N.: Network-based wireless location, IEEE Signal Process. Mag. 22(4):24–40
4. Alavi, B., Pahlavan, K.: Indoor geolocation distance error modeling with UWB channel measurements. In: Proceedings of the IEEE Personal Indoor Mobile Radio Communications Conference (PIMRC), Berlin, 11–14 September 2005
5. Bahl, P., Padmanabhan, V.N.: RADAR: An in-building RF-based user location and tracking system. Proceedings of the. IEEE INFOCOM 2000, Tel Aviv, Israel, 26–30 March 2000

6. Hightower, J., Borriello, G.: Location systems for ubiquitous computing. IEEE Comput. Mag. 34(8):57–66 (2001)
7. Niculescu, D.: Positioning in ad-hoc sensor networks. IEEE Network, 18(4):24–29
8. Kanaan, M., Pahlavan, K.: CN-TOAG: a new algorithm for indoor geolocation. In: Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Barcelona, 5–8 Sept 2004
9. Hatami, A., Pahlavan, K.: Comparative statistical analysis of indoor positioning using empirical data and indoor radio channel models. In: Proceedings of the Consumer Communications and Networking Conference, Las Vegas, Nevada, USA, 8–10 Jan 2006
10. Di Stefano, G., Graziosi, F., Santucci, F.: Distributed Positioning Algorithm for Ad-Hoc Networks. In: Proceedings of the IEEE International Workshop on UWB Systems, Oulu, Finnland, June 2003
11. Kanaan, M., Akgül, F.O., Alavi, B., Pahlavan, K.: Performance benchmarking of TOA-based UWB indoor geolocation systems using MSE profiling. In: Proceedings of the IEEE Vehicular Technology Conference, Montreal, Canada, 25–28 Sept 2006
12. US Federal Communications Commission: Revision of Part 15 of the Commission's Rules Regarding Ultra-Wideband Transmission Systems, FCC 02–48, First Report & Order, April 2004
13. McKelvin, M.L., Williams, M.L., Berry, N.M.: Integrated radio frequency identification and wireless sensor network architecture for automated inventory management and tracking applications. In: Proceedings of the Richard Tapia Celebration of Diversity in Computing Conference (TAPIA'05), Albuquerque, New Mexico, USA, 19–22 Oct 2005
14. Koo, S.G.M., Rosenberg, C., Chan, H-H, Lee, Y.C.: Location-based e-campus web services: from design to deployment. In: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, Fort Worth, Texas, USA 23–26 March 2003
15. Jensfelt, P.: Approaches to mobile robot localization in indoor environments. PhD thesis, Royal Institute of Technology, Stockholm (2001)
16. Ward, A., Jones, A., Hopper, A.: A new location technique for the active office. IEEE Personal Commun. Mag. 4(5):42–47
17. Pahlavan, K., Krishnamurthy, P., Hatami, A., Ylianttila, M., Mäkelä, J., Pichna, R., Vallström, J.: Handoff in hybrid mobile data networks. IEEE Personal Commun. Mag. **7**, 34–47 (2000)
18. Ko, Y., Vaidya, N.H.: Location-aided routing (LAR) in mobile ad hoc networks. In: Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking, 1998 (MOBICOM '98), Dallas, Texas, USA, 25–30 Oct 1998

Characteristic Travel Time

► Dynamic Travel Time Maps

Check-Out

► Smallworld Software Suite

Clementini Operators

► Dimensionally Extended Nine-Intersection Model (DE-9IM)

Cloaking Algorithms

CHI-YIN CHOW
Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA

Definition

Spatial cloaking is a technique used to blur a user's exact location into a spatial region in order to preserve her location privacy. The blurred spatial region must satisfy the user's specified privacy requirement. The most widely used privacy requirements are k-anonymity and minimum spatial area. The k-anonymity requirement guarantees that a user location is indistinguishable among k users. On the other hand, the minimum spatial area requirement guarantees that a user's exact location must be blurred into a spatial region with an area of at least $\mathcal{A}$, such that the probability of the user being located in any point within the spatial region is $1/\mathcal{A}$. A user location must be blurred by a spatial cloaking algorithm either on the client side or a trusted third-party before it is submitted to a location-based database server.

Main Text

This article surveys existing spatial cloaking techniques for preserving users' location privacy in location-based services (LBS) where users have to continuously report their locations to the database server in order to obtain the service. For example, a user asking about the nearest gas station has to report her exact location. With untrustworthy servers, reporting the location information may lead to several privacy threats. For example, an adversary may check a user's habit and interest by knowing the places she visits and the time of each visit. The key idea of a spatial cloaking algorithm is to perturb an exact user location into a spatial region that satisfies user specified privacy requirements, e. g., a k-anonymity requirement guarantees that a user is indistinguishable among k users.

Cross References

► Location-Based Services: Practices and Products
► Privacy Preservation of GPS Traces

Cloaking Algorithms for Location Privacy

CHI-YIN CHOW
Department of Computer Science and Engineering,
University of Minnesota, Minneapolis, MN, USA

Synonyms

Location blurring; Location perturbation; Location anonymization; Privacy; Location-privacy; Anonymity; Location-based services; Nearest neighbor; Peer to peer

Definition

Spatial cloaking is a technique to blur a user's exact location into a spatial region in order to preserve her location privacy. The blurred spatial region must satisfy the user's specified privacy requirement. The most widely used privacy requirements are k-anonymity and minimum spatial area. The k-anonymity requirement guarantees that a user location is indistinguishable among k users. On the other hand, the minimum spatial area requirement guarantees that a user's exact location must be blurred into a spatial region with an area of at least $\mathcal{A}$, such that the probability of the user being located in any point within the spatial region is $\frac{1}{\mathcal{A}}$. A user location must be blurred by a spatial cloaking algorithm either on the client side or a trusted third-party before it is submitted to a location-based database server.

Historical Background

The emergence of the state-of-the-art location-detection devices, e. g., cellular phones, global positioning system (GPS) devices, and radio-frequency identification (RFID) chips, has resulted in a location-dependent information access paradigm, known as location-based services (LBS). In LBS, mobile users have the ability to issue snapshot or continuous queries to the location-based database server. Examples of snapshot queries include "*where is the nearest gas station*" and "*what are the restaurants within one mile of my location*", while examples of continuous queries include "*where is the nearest police car for the next one hour*" and "*continuously report the taxis within one mile of my car location*". To obtain the precise answer of these queries, the user has to continuously provide her exact location information to a database server. With untrustworthy database servers, an adversary may access sensitive information about individuals based on their location information and queries. For example, an adversary may identify a user's habits and interests by knowing the places she visits and the time of each visit.

The k-anonymity model [12,13] has been widely used in maintaining privacy in databases [6,8,9,10]. The main idea is to have each tuple in the table as k-anonymous, i. e., indistinguishable among other $k-1$ tuples. However, none of these techniques can be applied to preserve user privacy for LBS, mainly for the reason that these approaches guarantee the k-anonymity for a snapshot of the database. In LBS, the user location is continuously changing. Such dynamic behavior requires continuous maintenance of the k-anonymity model. In LBS, k-anonymity is a user specified privacy requirement which may have a different value for each user.

Scientific Fundamentals

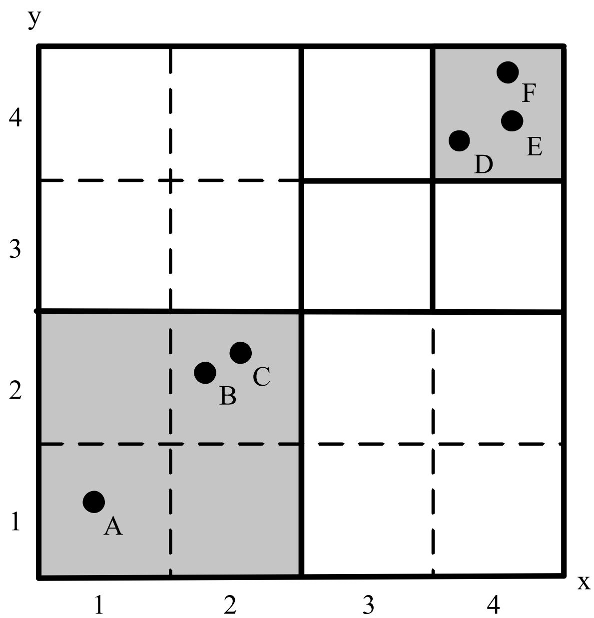
Spatial cloaking algorithms can be divided into two major types: k-anonymity spatial cloaking [2,3,4,5,7,11] and uncertainty spatial cloaking [1]. k-anonymity spatial cloaking aims to blur user locations into spatial regions which satisfy the user's specified k-anonymity requirement, while uncertainty spatial cloaking aims to blur user locations into spatial regions which stratify the user's specified minimum spatial area requirement.

Adaptive Interval Cloaking

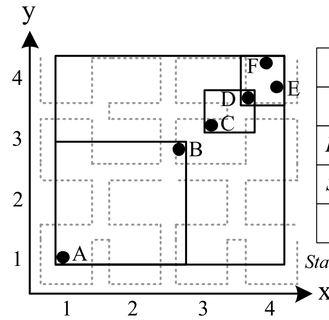
This approach assumes that all users have the same k-anonymity requirements [3]. For each user location update, the spatial space is recursively divided in a KD-tree-like format until a minimum k-anonymous subspace is found. Such a technique lacks scalability as it deals with each single movement of each user individually. Figure 1 depicts an example of the adaptive interval cloaking algorithm in which the k-anonymity requirement is three. If the algorithm wants to cloak user A's location, the system space is first divided into four equal subspaces, $\langle$(1,1), (2,2)$\rangle$, $\langle$(3,1), (4,2)$\rangle$, $\langle$(1,3), (2,4)$\rangle$, and $\langle$(3,3), (4,4)$\rangle$. Since user A is located in the subspaces $\langle$(1,1), (2,2)$\rangle$, which contains at least k users, these subspaces are further divided into four equal subspaces, $\langle$(1,1), (1,1)$\rangle$, $\langle$(2,1), (2,1)$\rangle$, $\langle$(1,2), (1,2)$\rangle$, and $\langle$(2,2), (2,2)$\rangle$. However, the subspace containing user A does not have at least k users, so the minimum suitable subspace is $\langle$(1,1), (2,2)$\rangle$. Since there are three users, D, E, and F, located in the cell (4,4), this cell is the cloaked spatial region of their locations.

CliqueCloak

This algorithm assumes a different k-anonymity requirement for each user [3]. CliqueCloak constructs a graph and cloaks user locations when a set of users forms a clique in the graph. All users share the same cloaked spatial region which is a minimum bounding box covering them. Then,



Cloaking Algorithms for Location Privacy, Figure 1 Adaptive interval cloaking ($k = 3$)



Users	A	B	C	D	E	F
k_u	6	2	2	3	3	3
Rank(u)	0	1	2	3	4	5
Start(u)	0	0	2	3	3	3
End(u)	5	1	3	5	5	5

$Start(u) = Rank(u) - (Rank(u) \bmod k_u)$

$End(u) = Start(u) + k_u - 1$

Cloaking Algorithms for Location Privacy, Figure 2 hilbASR

the cloaked spatial region is reported to a location-based database server as their locations. Users can also specify the maximum area of the cloaked region which is considered as a constraint on the clique graph, i. e., the cloaked spatial region cannot be larger than the user's specified maximum acceptable area.

k-Area Cloaking

This scheme keeps suppressing a user location into a region which covers at least $k - 1$ other sensitive areas, e. g., restaurants, hospitals, and cinemas around the user's current sensitive area [5]. Thus, the user resident area is indistinguishable among k sensitive areas. This spatial cloaking algorithm is based on a map which is partitioned into zones, and each zone contains at least k sensitive areas. Thus, the continuous movement of users is just abstracted as moving between zones. Users can specify their own privacy requirements by generalizing personalized sensitivity maps.

Hilbert *k*-Anonymizing Spatial Region (hilbASR)

Here, users are grouped together into variant buckets based on the Hilbert ordering of user locations and their own k-anonymity requirements [7]. Using the dynamic hilbASR, the cloaked spatial regions of users A to F can be determined by using two equations, *start*(u) and *end*(u), which are depicted in Fig. 2, where *start*(u) and *end*(u) indicate the start and end rankings of a cloaked spatial region, respectively, u is a user identity, and the dotted line represents the Hilbert ordering.

Nearest-Neighbor *k*-Anonymizing Spatial Region (nnASR)

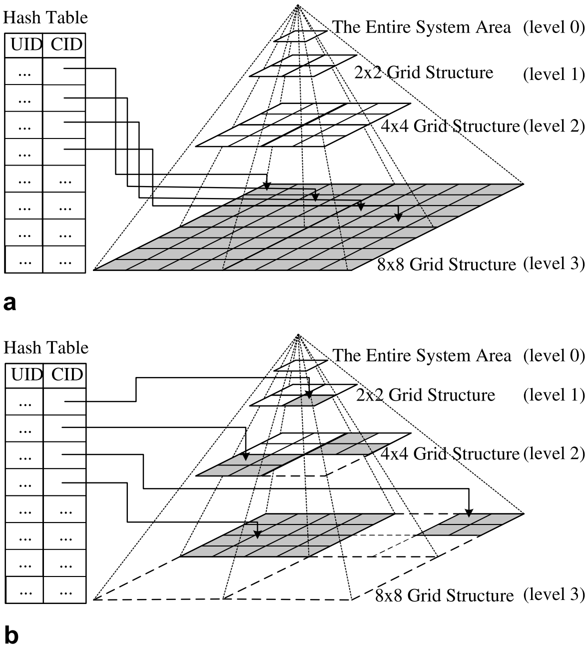
This is the randomized version of a k-nearest neighbor scheme [7]. For a user location u, the algorithm first determines a set S of k-nearest neighbors of u, including u. From S, the algorithm selects a random user u' and forms a new set S' that includes u' and the $k - 1$ nearest neighbors of u'. Then, another new set S'' is formed by taking a union between S and S''. Finally, the required cloaked spatial region is the bounding rectangle or circle which covers all the users of S''.

Uncertainty

This approach proposes two uncertainty spatial cloaking schemes, *uncertainty region* and *coverage of sensitive area* [1]. The uncertainty region scheme simply blurs a user location into an uncertainty region at a particular time t, denoted as $U(t)$. The larger region size means a more strict privacy requirement. The coverage of sensitive area scheme is proposed for preserving the location privacy of users who are located in a sensitive area, e. g., hospital or home. The coverage of sensitive area for a user is defined as Coverage $= \frac{\text{Area(sensitive area)}}{\text{Area(uncertainty region)}}$. The lower value of the coverage indicates a more strict privacy requirement.

Casper

Casper supports both the k-anonymity and minimum spatial area requirements [11]. System users can dynamically change their own privacy requirements at any instant. It proposes two grid-based pyramid structures to improve system scalability, *complete pyramid* and *incomplete pyramid*.



Cloaking Algorithms for Location Privacy, Figure 3 Grid-based pyramid data structures. **a** Complete pyramid. **b** Incomplete pyramid

Complete Pyramid Figure 3a depicts the complete pyramid data structure which hierarchically decomposes the spatial space into H levels where a level of height h has 4^h grid cells. The root of the pyramid is of height zero and has only one grid cell that covers the whole space. Each pyramid cell is represented as (cid, N), where cid is the cell identifier and N is the number of mobile users within the cell boundaries. The pyramid structure is dynamically maintained to keep track of the current number of mobile users within each cell. In addition, the algorithm keeps track of a hash table that has one entry for each registered mobile user with the form (*uid*, *profile*, *cid*), where *oid* is the mobile user identifier, *profile* contains the user specified privacy requirement, and *cid* is the cell identifier in which the mobile user is located. The *cid* is always in the lowest level of the pyramid (the shaded level in Fig. 3a).

Incomplete Pyramid The main idea of the incomplete pyramid structure is that not all grid cells are appropriately maintained. The shaded cells in Fig. 3b indicate the lowest level cells that are maintained.

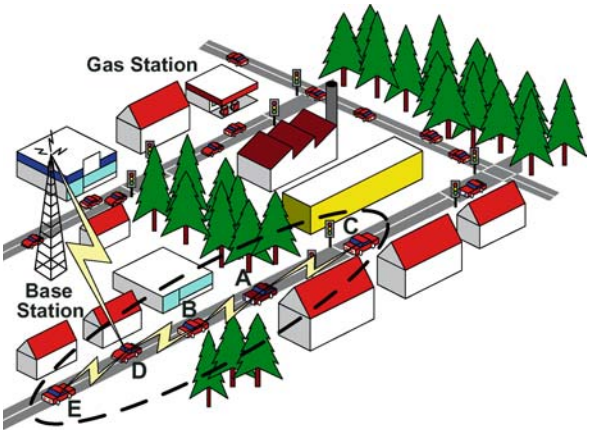
Cloaking Algorithm Casper adopts a bottom-up cloaking algorithm which starts at a cell where the user is located at from the lowest maintained level and then traverses up the pyramid structure until a cell satisfying the user specified privacy requirement is found. The resulting cell is used as the cloaked spatial region of the user location. In addition to the regular maintenance procedures as that of the basic location anonymizer, the adaptive location anonymizer is also responsible for maintaining the shape of the incomplete pyramid. Due to the highly dynamic environment, the shape of the incomplete pyramid may have frequent changes. Two main operations are identified in order to maintain the efficiency of the incomplete pyramid structure, namely, *cell splitting* and *cell merging*.

In the cell splitting operation, a cell *cid* at level i needs to be split into four cells at level $i+1$ if there is at least one user u in *cid* with a privacy profile that can be satisfied by some cell at level $i+1$. To maintain such criterion, Casper keeps track of the most relaxed user u_r for each cell. If a newly coming object u_{new} to the cell *cid* has a more relaxed privacy requirement than u_r, the algorithm checks if splitting cell *cid* into four cells at level $i+1$ would result in having a new cell that satisfies the privacy requirements of u_{new}. If this is the case, the algorithm will split cell *cid* and distribute all its contents to the four new cells. However, if this is not the case, the algorithm just updates the information of u_r. In case one of the users leaves cell *cid*, the algorithm will just update u_r if necessary.

In the cell merging operation, four cells at level i are merged into one cell at a higher level $i-1$ only if all the users in the level i cells have strict privacy requirements that cannot be satisfied within level i. To maintain this criterion, the algorithm keeps track of the most relaxed user u'_r for the four cells of level i together. If such a user leaves these cells, the algorithm has to check upon all existing users and make sure that they still need cells at level i. If this is the case, the algorithm just updates the new information of u'_r. However, if there is no need for any cell at level i, the algorithm merges the four cells together into their parent cell. In the case of a new user entering cells at level i, the algorithm just updates the information of u'_r if necessary.

Peer-to-Peer Spatial Cloaking

This algorithm also supports both the k-anonymity and minimum spatial area requirements [2]. The main idea is that before requesting any location-based service, the mobile user will form a group from her peers via single-hop and/or multi-hop communication. Then, the spatial cloaked area is computed as the region that covers the entire group of peers. Figure 4 gives an illustrative example of peer-to-peer spatial cloaking. The mobile user A wants to find her nearest gas station while being five anonymous, i. e., the user is indistinguishable among five users. Thus, the mobile user A has to look around and find four other peers to collaborate as a group. In this example, the four



Cloaking Algorithms for Location Privacy, Figure 4 An example of peer-to-peer spatial cloaking

peers are B, C, D, and E. Then, the mobile user A cloaks her exact location into a spatial region that covers the entire group of mobile users A, B, C, D, and E. The mobile user A randomly selects one of the mobile users within the group as an *agent*. In the example given in Fig. 4, the mobile user D is selected as an agent. Then, the mobile user A sends her query (i. e., what is the nearest gas station) along with her cloaked spatial region to the agent. The agent forwards the query to the location-based database server through a base station. Since the location-based database server processes the query based on the cloaked spatial region, it can only give a list of candidate answers that includes the actual answers and some false positives. After the agent receives the candidate answers, it forwards the candidate answers to the mobile user A. Finally, the mobile user A gets the actual answer by filtering out all the false positives.

Key Applications

Spatial cloaking techniques are mainly used to preserve location privacy, but they can be used in a variety of applications.

Location-Based Services

Spatial cloaking techniques have been widely adopted to blur user location information before it is submitted to the location-based database server, in order to preserve user location privacy in LBS.

Spatial Database

Spatial cloaking techniques can be used to deal with some specific spatial queries. For example, given an object location, find the minimum area which covers the object and other $k-1$ objects.

Data Mining

To perform data mining on spatial data, spatial cloaking techniques can be used to perturb individual location information into lower resolution to preserve their privacy.

Sensor-Based Monitoring System

Wireless sensor networks (WSNs) promise to have a vast significant academic and commercial impact by providing real-time and automatic data collection, monitoring applications and object positioning. Although sensor-based monitoring or positioning systems clearly offer convenience, the majority of people are not convinced to use such systems because of privacy issues. To overcome this problem, an in-network spatial cloaking algorithm can be used to blur user locations into spatial regions which satisfy user specified privacy requirements before location information is sent to a sink or base station.

Future Directions

Existing spatial cloaking algorithms have limited applicability as they are: (a) *applicable only for snapshot locations and queries.* As location-based environments are characterized by the *continuous* movements of mobile users, spatial cloaking techniques should allow continuous privacy preservation for both user locations and queries. Currently, existing spatial cloaking algorithms only support snapshot location and queries. (b) *not distinguishing between location and query privacy*. In many applications, mobile users do not mind that their exact location information is revealed, however, they would like to hide the fact that they issue some location-based queries as these queries may reveal their personal interests. Thus far, none of the existing spatial cloaking algorithms support such a relaxed privacy notion where it is always assumed that users have to hide both their locations and the queries they issue. Examples of applications that call for such a new relaxed notion of privacy include: (1) *Business operation*. A courier business company has to know the location of its employees in order to decide which employee is the nearest one to collect a certain package. However, the company is not allowed to keep track of the employees' behavior in terms of their location-based queries. Thus, company employees reveal their location information, but not their query information. (2) *Monitoring system*. Monitoring systems (e.g, transportation monitoring) rely on the accuracy of user locations to provide their valuable services. In order to convince users to participate in these systems, certain privacy guarantees should be imposed on their behavior through guaranteeing the privacy of their location-based queries even though their locations will be revealed.

Cross References

▶ Location-Based Services: Practices and Products
▶ Privacy Preservation of GPS Traces
▶ Privacy and Security Challenges in GIS

Recommended Reading

1. Cheng, R., Zhang, Y., Bertino, E., Prabhakar, S.: Preserving User Location Privacy in Mobile Data Management Infrastructures. In: Proceedings of Privacy Enhancing Technology Workshop. pp. 393–412. Cambridge, UK (2006)
2. Chow, C.Y., Mokbel, M.F., Liu, X.: A Peer-to-Peer Spatial Cloaking Algorithm for Anonymous Location-based Services. In: Proceedings of the ACM Symposium on Advances in Geographic Information Systems (ACM GIS), pp. 171–178. Arlington, Virginia, USA (2006)
3. Gedik, B., Liu, L.: A Customizable *k*-Anonymity Model for Protecting Location Privacy. In: Proceedings of the International Conference on Distributed Computing Systems (ICDCS), pp. 620–629. Columbus, OH, USA (2005)
4. Gruteser, M., Grunwald, D.: Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In: Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys), pp. 31–42. San Francisco, CA, USA (2003)
5. Gruteser, M., Liu, X.: Protecting Privacy in Continuous Location-Tracking Applications. IEEE Secur. Priv. **2**(2), 28–34 (2004)
6. Bayardo Jr., R.J., Agrawal, R.: Data Privacy through Optimal *k*-Anonymization. In: Proceedings of the International Conference on Data Engineering (ICDE), pp. 217–228. Tokyo, Japan (2005)
7. Kalnis, P., Ghinita, G., Mouratidis, K., Papadias, D.: Preserving Anonymity in Location Based Services. Technical Report TRB6/06, Department of Computer Science, National University of Singapore (2006)
8. LeFevre, K., DeWitt, D., Ramakrishnan, R.: Mondrian Multidimensional *k*-Anonymity. In: Proceedings of the International Conference on Data Engineering (ICDE), Atlanta, GA, USA (2006)
9. LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Incognito: Efficient Full-Domain *k*-Anonymity. In: Proceedings of the ACM International Conference on Management of Data (SIGMOD), pp. 29–60. Baltimore, Maryland, USA (2005)
10. Meyerson, A., Williams, R.: On the Complexity of Optimal *K*-Anonymity. In: Proceedings of the ACM Symposium on Principles of Database Systems (PODS), pp.223–228. Paris, France (2004)
11. Mokbel, M.F., Chow, C.Y., Aref, W.G.: The New Casper: Query Processing for Location Services without Compromising Privacy. In: Proceedings of the International Conference on Very Large Data Bases (VLDB), pp. 763–774, Seoul, Korea (2006)
12. Sweeney, L.: Achieving *k*-anonymity Privacy Protection using Generalization and Suppression. Int. J. Uncertain. Fuzziness Knowl. Sys. **10**(5), 571–588 (2002)
13. Sweeney, L.: *k*-anonymity: A Model for Protecting Privacy. Intern. J. Uncertain., Fuzziness Knowl.-based Syst. **10**(5), 557–570 (2002)

Close Range

▶ Photogrammetric Applications

Closest Point Query

▶ Nearest Neighbor Query

Closest Topological Distance

▶ Conceptual Neighborhood

Cluster Analysis

▶ Geodemographic Segmentation

Cognition

▶ Hierarchies and Level of Detail

Cognitive Engineering

▶ Geospatial Semantic Web: Personalisation

Cognitive Mapping

▶ Wayfinding, Landmarks

Cognitive Psychology

▶ Wayfinding: Affordances and Agent Simulation

Collaborative Geographic Information Systems

▶ Geocollaboration

Co-location

▶ Patterns, Complex

Co-location Mining

▶ Co-location Pattern Discovery

Collocation Pattern

▶ Co-location Pattern

Co-location Pattern

NIKOS MAMOULIS
Department of Computer Science,
University of Hong Kong, Hong Kong, China

Synonyms

Spatial association pattern; Collocation pattern

Definition

A (spatial) *co-location pattern* P can be modeled by an undirected connected graph where each node corresponds to a non-spatial feature and each edge corresponds to a neighborhood relationship between the corresponding features. For example, consider a pattern with three nodes labeled "timetabling", "weather", and "ticketing", and two edges connecting "timetabling" with "weather" and "timetabling" with "ticketing". An *instance* of a pattern P is a set of objects that satisfy the unary (feature) and binary (neighborhood) constraints specified by the pattern's graph. An instance of an example pattern is a set $\{o_1, o_2, o_3\}$ of three spatial locations where $label(o_1) =$ "timetabling", $label(o_2) =$ "weather", $label(o_3) =$ "ticketing" (unary constraints) and $dist(o_1, o_2) \leq \varepsilon$, $dist(o_1, o_3) \leq \varepsilon$ (spatial binary constraints). In general, there may be an arbitrary spatial (or spatio-temporal) constraint specified at each edge of a pattern graph (for example, topological, distance, direction, and time-difference constraints).

Main Text

Co-location patterns are used to derive co-location rules that associate the existence of non-spatial features in the same spatial neighborhood. An example of such a rule is "if a water reservoir is contaminated, then people who live in nearby houses have high probability of having a stomach disease". The interestingness of a co-location pattern is quantized by two measures; the *prevalence* and the *confidence*. Co-location patterns can be mined from large spatial databases with the use of algorithms that combine (multi-way) spatial join algorithms with spatial association rule mining techniques.

Cross References

► Patterns, Complex
► Retrieval Algorithms, Spatial

Co-location Pattern Discovery

WEI HU
International Business Machines Corp.,
Rochester, MN, USA

Synonyms

Co-location rule finding; Co-location mining; Co-location rule mining; Co-location rule discovery; Co-occurrence; Spatial association; Spatial association analysis

Definition

Spatial co-location rule discovery or *spatial co-location pattern discovery* is the process that identifies spatial co-location patterns from large spatial datasets with a large number of Boolean spatial features.

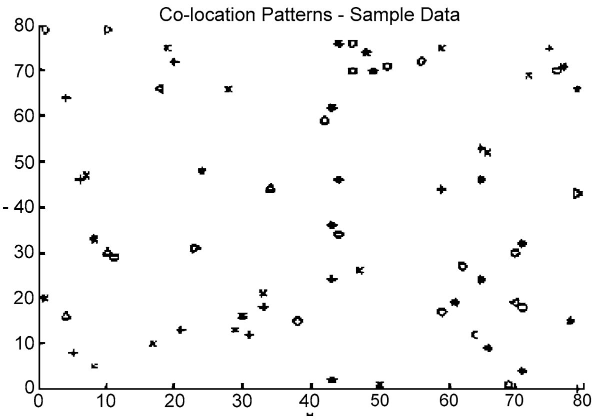
Historical Background

The co-location pattern and rule discovery are part of the spatial data mining process. The differences between spatial data mining and classical data mining are mainly related to data input, statistical foundation, output patterns, and computational process. The research accomplishments in this field are primarily focused on the output pattern category, specifically the predictive models, spatial outliers, spatial co-location rules, and clusters [1].

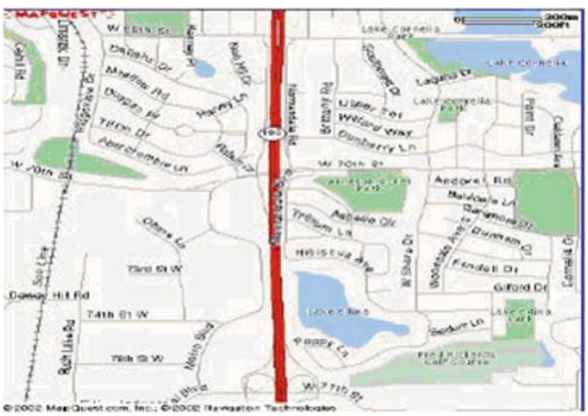
The spatial pattern recognition research presented here, which is focused on co-location, is also most commonly referred to as the spatial co-location pattern discovery and co-location rule discovery. To understand the concepts of spatial co-location pattern discovery and rule discovery, we will have to first examine a few basic concepts in spatial data mining.

The first word to be defined is Boolean spatial features. *Boolean spatial features* are geographic object types. They either are absent or present regarding different locations within the domain of a two dimensional or higher (three) dimensional metric space such as the surface of the earth [1]. Some examples of Boolean spatial features are categorizations such as plant species, animal species, and types of roads, cancers, crimes and business.

The next concept relates to co-location patterns and rules. *Spatial co-location patterns* represent the subsets of Boolean spatial features whose instances are often located in close geographic proximity [1]. It resembles frequent patterns in many aspects. Good examples are symbiotic species. The Nile crocodile and Egyptian plover in ecology prediction (Fig. 1) is one good illustration of a point spatial co-location pattern representation. Frontage roads



Co-location Pattern Discovery, Figure 1 Illustration of point spatial co-location patterns. Shapes represent different spatial feature types. Spatial features in sets {'+', 'x'} and {'o', '*'} tend to be located together [1]



Co-location Pattern Discovery, Figure 2 Illustration of Line String Co-location Patterns. Highways, e. g., Hwy100, and frontage roads, e. g., Normandale Road, are co-located [1]

and highways (Fig. 2) in specified metropolitan road maps could be used to demonstrate line-string co-location patterns.

Examples of various categories of spatial co-location patterns are given in Table 1. We can see that the domains of co-location patterns are distributed in many interesting fields of science research and daily services, which proves their great usefulness and importance.

Spatial co-location rules are models to associate the presence of known Boolean spatial features referencing the existence of instances of other Boolean spatial features in the neighborhood. Figure 1 also provides good examples of spatial co-location rules. As can be seen, rule "Nile crocodiles → Egyptian plover" can predict the presence of Egyptian plover birds in the same areas where Nile crocodiles live. A dataset consisting of several different Boolean spatial feature instances is marked on the space. Each type of Boolean spatial features is distinguished by a distinct representation shape. A careful examination reveals two co-location patterns: ('+', 'x') and ('o', '*') [1]. Spatial co-location rules can be further classified into popular rules and confident rules, according to the frequency of cases showing in the dataset. The major concern here is the difference of dealing with rare events and popular events. Usually, rare events are ignored, and only the popular co-location rules are mined. So if there is a need to identify the confident co-location rules, then special handling and a different approach must be taken to reach them [3].

Spatial co-location rule discovery is the process that identifies spatial co-location patterns from large spatial datasets with a large number of Boolean spatial features [1]. The problems of spatial co-location rule discovery are similar to the *spatial association rule mining* problem, which identifies the inter-relationships or associations among a number of spatial datasets. The difference between the two has to do with the concept of transactions.

An example of association rule discovery can be seen with market basket datasets, in which transactions represent sets of merchandise item categories purchased all-together by customers [1]. The association rules are derived from all the associations in the data with support values that exceed a user defined threshold. In this example, we can define in detail the process of mining association rules as to identify frequent item sets in order to plan store layouts or marketing campaigns as a part of related business intelligence analysis.

On the other hand, in a spatial co-location rule discovery problem, we usually see that the transactions are not explicit [1]. There are no dependencies among the transactions analyzed in market basket data, because the transaction data do not share instances of merchandise item categories but rather instances of Boolean spatial features instead. These Boolean spatial features are distributed into a continuous space domain and thus share varied spatial types of relationships, such as overlap, neighbor, etc. with each other.

Although spatial co-location patterns and co-location rules differ slightly, according to the previous definitions, it can be said that *spatial co-location pattern discovery* is merely another phrasing for spatial co-location rule finding. Basically, the two processes are the same and can be used in place of each other. Both are used to find the frequent co-occurrences among Boolean spatial features from given datasets.

Co-location Pattern Discovery, Table 1 Examples of co-location patterns [2]

Domains	Example features	Example co-location patterns
Ecology	Species	Nile crocodile, Egyptian plover
Earth science	Climate and disturbance events	Wild fire, hot, dry, lightning
Economics	Industry types	Suppliers, producers, consultants
Epidemiology	Disease types and environmental events	West Nile disease, stagnant water sources, dead birds, mosquitoes
Location-based service	Service type requests	Tow, police, ambulance
Weather	Fronts, precipitation	Cold front, warm front, snow fall
Transportation	Delivery service tracks	US Postal Service, UPS, newspaper delivery

Scientific Fundamentals

According to one categorization, there are three methods of finding co-location patterns in spatial datasets, depending on the focus of the search. These three categories are the *reference feature centric model*, the *window centric model* and the *event centric model* [4]. The *reference feature centric model* is relevant to application domains that focus on a specific Boolean spatial feature such as cancer. The goal of the scientists is to find the colocation patterns between this Boolean spatial feature and other task related features such as asbestos or other substances. This model uses the concept of neighborhood relationship to materialize the transactions from datasets. Measurements of support and confidence can be used to show the degree of interestingness [4]. For example, if there are two features A and B, and if A is the relevant feature, then B is said to be close to A if B is a neighbor of A. But how can we tell that B is a neighbor of A? Here we can use either the Euclidean distance or the Manhattan distance, depending on the type of application domain we are investigating. Then with the corresponding definition of the distance between the features, we could declare them to be neighbors. Thus by considering A, all the other Boolean spatial features surrounding A are used as transactions. Once the data is materialized as above, the support and confidence are computed and used to measure degree of interestingness [4]. Table 2 shows an instance of data and Fig. 3 illustrates the layout and process we described with detailed data from Table 2.

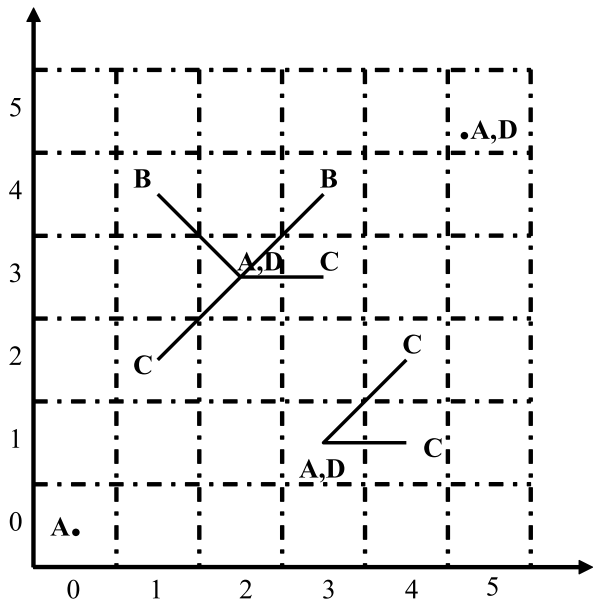
The second method of finding co-location patterns is a *window centric model*, or *data partitioning model*. The process defines proper sized windows and then enumerates all possible windows as transactions. Each window is actually a partition of the whole space and the focus is on the local co-location patterns, which are bounded by the window boundaries. Patterns across multiple windows are of no concern. Each window is a transaction and the process tries to find which features appear together the most number of times in these transactions, alias, windows, i. e., using support and confidence measurements [4].

Figure 4 shows the processing with window partitions on data similar to that shown in Fig. 3. As this is a local model, even though here the A and C could have been a pattern, these features are completely ignored since they are not within a single window.

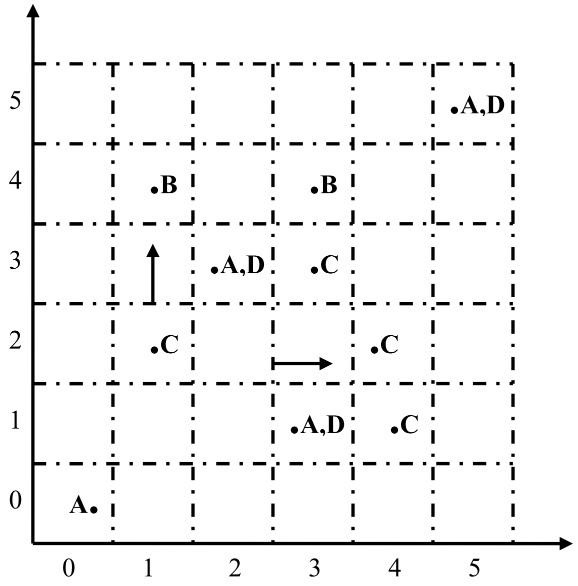
The third modeling method is the *event centric model*. This model is mostly related to ecology specific domains

Instance of A	Transaction
(0,0)	∅
(2,3)	{B,C}
(3,1)	{C}
(5,5)	∅

Co-location Pattern Discovery, Table 2 Boolean feature A and the defined transactions related to B and C



Co-location Pattern Discovery, Figure 3 Transactions are defined around instances of feature A, relevant to B and C [4]



Co-location Pattern Discovery, Figure 4 Example of window centric model [4]

where scientists want to investigate specific events such as drought, El Nino etc. The goal of this model is to find the subsets of spatial features likely to occur in the neighborhood of a given event type. One of the assumptions of this algorithm is that the neighbors are reflexive, that is, interchangeable. For example, if A is a neighbor of B, then B is also a neighbor of A.

The event centric defines key concepts as follows: "A *neighborhood* of l is a set of locations L = {l1, l2, l3, ...,lk} such that li is a neighbor of l" [4]. "I = {I1,...,Ik} is a *row instance* of a co-location C = {f1,...,fk} if Ij is an instance of feature fj" [4]. The *participation ratio* and *participation index* are two measures which replace support and confidence here. The participation ratio is the number of row instances of co-location C divided by number of instances of Fi. Figure 5 shows an example of this model.

Table 3 shows a summary of the interest measures for the three different models.

With different models to investigate different problems of various application domains, there are also multiple algorithms used in the discovery process. Approaches to discover co-location rules can be categorized into two classes, *spatial statistics* and *data mining approaches*.

Spatial statistics-based approaches use measures of spatial correlation to characterize the relationship between different types of spatial features. Measures of spatial correlation include the cross-K function with Monte Carlo simulation, mean nearest-neighbor distance, and spatial regression models. Computing spatial correlation measures for all possible co-location patterns can be computationally expensive due to the exponential number of candidate subsets extracted from a large collection of spatial Boolean features that we are interested in [6].

Data mining approaches can be further divided into two categories: the *clustering-based map overlay approach* and the *association rule-based approaches*.

Clustering-based map overlay approach regards every spatial attribute as a map layer and considers spatial clusters (regions) of point-data in each layer as candidates for mining the associations among them. *Association rule-based approaches* again can be further divided into two categories: the *transaction-based approaches* and the *distance-based approaches*.

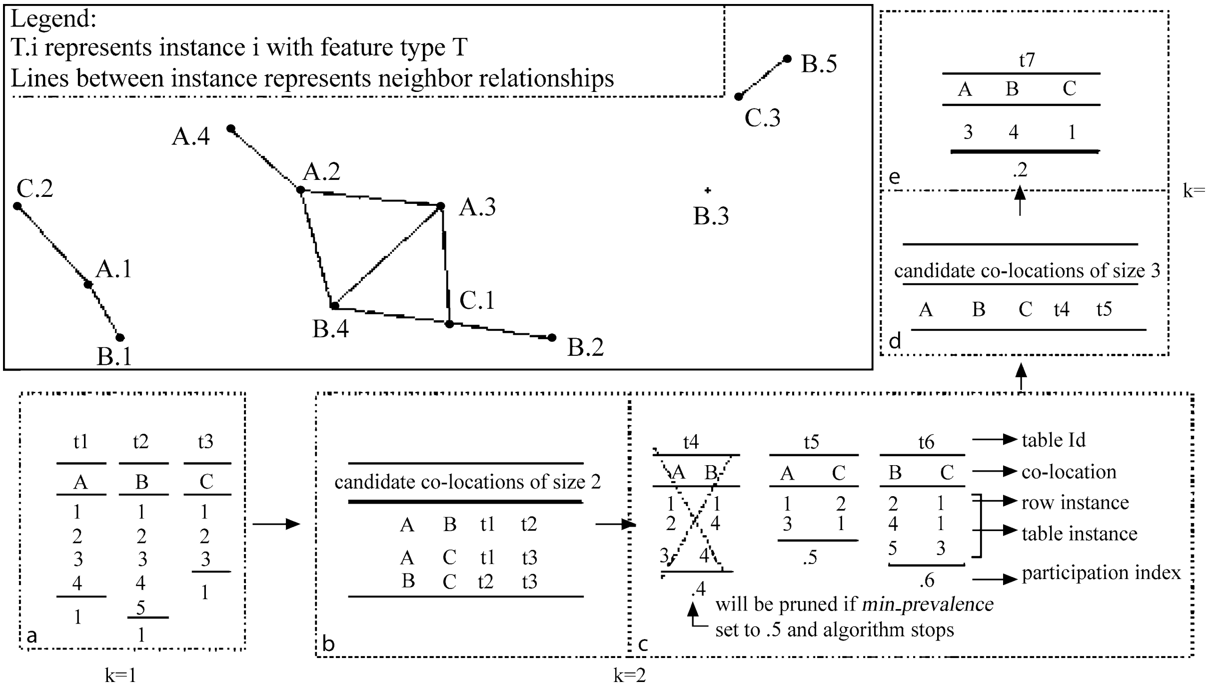
Transaction-based approaches aim to define transactions over space such that an A priori-like algorithm can be used just as in the association rule discovery process. Transactions over space can be defined by a reference centric model as discussed previously, which enables the derivation of association rules using the A priori algorithm. There are few major shortcomings of this approach: generalization of this paradigm is non-trivial in the case where no reference feature is specified; and duplicate counts for many candidate associations may result when defining transactions around locations of instances of all features.

Distance-based approaches are relatively novel. A couple of different approaches have been presented by different research groups. One proposes the participation index as the prevalence measure, which possesses a desirable anti-monotone property [3]. Thus a unique subset of co-location patterns can be specified with a threshold on the participation index without consideration of detailed algorithm applied such as the order of examination of instances of a co-location. Another advantage of using the participation index is that it can define the correctness and completeness of co-location mining algorithms.

Key Applications

The problem of mining spatial co-location patterns can be applied to many useful science-research or public interest domains.

As shown in Table 1, one of the top application domains is location based services. With advances such as GPS and mobile communication devices, many location based services have been introduced to fulfill users' increasing desires for convenience. Many of the services requested by service subscribers from their mobile devices see benefit from the support of spatial co-location pattern mining. The location-based service provider needs to know which



Co-location Pattern Discovery, Figure 5 Event centric model example [6]

Co-location Pattern Discovery, Table 3 Interest measures for different models [1]

Model	Items	Transactions defined by	Interest measures for $C_1 \to C_2$	
			Prevalence	Conditional probability
Reference feature centric	Predicates on reference and relevant features	Instances of reference feature C_1 and C_2 involved with	Fraction of instance of reference feature with $C_1 \cup C_2$	Pr(C_2 is true for an instance of reference features given C_1 is true for that instance of reference feature)
Data partitioning	Boolean feature types	A partitioning of spatial dataset	Fraction of partitions with $C_1 \cup C_2$	Pr(C_2 in a partition given C_1 in that partition)
Event centric	Boolean feature types	Neighborhoods of instances of feature types	Participation index of $C_1 \cup C_2$	Pr(C_2 in a neighborhood of C_1)

requests are submitted frequently together and which are located in spatial proximity [2].

Ecology is another good field to apply this technology. Because ecologists are very interested in finding frequent co-occurrences among spatial features, such as drought, EI Nino, substantial increase/drop in vegetation, and extremely high precipitation [2].

A third important domain whose future cannot be imagined without spatial data mining is weather services. The identification of correct and valuable co-location patterns or rules from huge amounts of colleted historical data can be expected to lead to better predictions about incoming weather, deeper insights into environmental impacts on weather patterns, and suggestions of possible effective steps to prevent the future deterioration of the environment.

A final example in our list of applications is traffic control or transportation management. With the knowledge of co-location rules discovered from existing datasets, better supervising and management could be carried out to make transportation systems run in the most efficient way, as well as to gain clearer foresights of future road network development and expansion.

There are many more interesting fields related to the spatial co-location application domain, such as disease research, economics, earth science, etc. [5]. With the availability of more spatial data from different areas, we can expect more research and studies to benefit from this technology.

Future Directions

Spatial co-location pattern discovery and co-location rule mining are very important, even essential tasks of a *spatial data mining systems* (SDMS), which extract previously unknown but interesting spatial patterns and relationships from large spatial datasets. These methods have the potential to serve multiple application domains and have a wide impact on many scientific research fields and services. Current approaches to mine useful *co-location patterns* are still evolving with new studies carried out in the field. We can expect extended development of such techniques to improve the algorithms and efficiency in future studies.

A first potential direction of research is to find more efficient algorithms against extended spatial data types other than points, such as line segments and polygons [6].

Second direction is that as only *Boolean spatial features* are mined here, the future studies can extend the *co-location mining* framework to handle categorical and continuous features that also exist in the real world [6].

Third potential extension can be on the notion of co-location pattern to be de-colocation pattern or co-incidence pattern [2].

Recommended Reading

1. Shekhar, S., Zhang, P., Huang, Y., Vatsavai, R.R.: Trends in Spatial Data Mining. In: Kargupta, H., Joshi, A., Sivakumar, K., Yesha, Y. (eds.) Data Mining: Next Generation Challenges and Future Directions. AAAI/MIT Press, Cambridge, MA (2003)
2. Xiong, H., Shekhar, S., Huang, Y., Kumar, V., Ma, X., Yoo J.: A framework for discovering co-location patterns in data sets with extended spatial objects. In Proc. of SIAM International Conference on Data Mining (SDM) (2004)
3. Huang, Y., Xiong, H., Shekhar, S., Pei, J.: Mining confident colocation rules without a support threshold. In: Proc. of the 18th ACM Symposium on Applied Computing (ACM SAC), Melbourne, FL March 2003
4. Shekhar, S., Huang, Y.: Discovering spatial co-location patterns: a summary of results. In: Proc. of 7th Intl. Symp. on Spatial and Temporal Databases (SSTD), Redondo Beach, CA July 2001
5. Shekhar, S., Schrater, P., Raju, W., Wu, W.: Spatial contextual classification and prediction models for mining geospatial data. IEEE Transactions on Multimedia (2002)
6. Huang, Y., Shekhar, S., Xiong, H.: Discovering co-location patterns from spatial datasets: A general approach. IEEE Transactions on Knowledge and Data Engineering (TKDE), 16, NO. 12 December 2004

Co-location Patterns

► Co-location Patterns, Interestingness Measures

Co-location Patterns, Algorithms

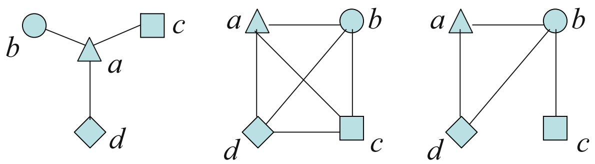
NIKOS MAMOULIS
Department of Computer Science,
University of Hong Kong, Hong Kong, China

Synonyms

Mining collocation patterns; Mining spatial association patterns; Co-occurrence; Association; Participation ratio; Participation index; Reference-feature centric

Definition

A spatial co-location pattern associates the co-existence of a set of non-spatial features in a spatial neighborhood. For example, a co-location pattern can associate contaminated water reservoirs with a certain disease within 5 km distance from them. For a concrete definition of the problem, consider number n of spatial datasets $R_1, R_2, \ldots, R_n$, such that each R_i contains objects that have a common non-spatial feature f_i. For instance, R_1 may store locations of water sources, R_2 may store locations of appearing disease symptoms, etc. Given a distance threshold ε, two objects on the map (independent of their feature labels) are *neighbors* if their distance is at most ε. We can define a *co-location pattern* P by an undirected connected graph where each node corresponds to a feature and each edge corresponds to a neighborhood relationship between the corresponding features. Figure 1 shows examples of a *star* pattern, a *clique* pattern and a generic one. A variable labeled with feature f_i is only allowed to take instances of that feature as values. Variable pairs that should satisfy a spatial relationship (i. e., constraint) in a valid pattern instance are linked by an edge. In the representations of Fig. 1, we assume that there is a single constraint type (e. g., close to), however in the general case, any spatial relationship could label each edge. Moreover, in the general case, a feature can label more than two variables. Patterns with more than one variable of the same label can be used to describe *spatial autocorrelations* on a map.



a star **b** clique **c** generic

Co-location Patterns, Algorithms, Figure 1 Three pattern representations

Interestingness measures [3,10] for co-location patterns express the statistical significance of their instances. They can assist the derivation of useful rules that associate the instances of the features.

Historical Background

The problem of mining association rules based on spatial relationships (e. g., adjacency, proximity, etc.) of events or objects was first discussed in [4]. The spatial data are converted to transactional data according to a *reference feature* model. Later, the research interest shifted toward mining *co-location patterns*, which are feature centric sets with instances that are located in the same neighborhood [3,6,7,10,14]. [3,6,7,10] focused on patterns where the closeness relationships between features form a complete graph (i. e., every pair of features should be close to each other in a pattern), whereas [14] extended this model to feature-sets with closeness relationships between arbitrary pairs and proposed an efficient algorithm for mining such patterns (which is herein reviewed). [12] extended the concept of co-locations for objects with extend and shape, whereas [11] studied the mining of co-location patterns that involve spatio-temporal topological constraints.

Scientific Fundamentals

Consider a number n of spatial datasets $R_1, R_2, \ldots, R_n$, such that each R_i contains all objects that have a particular non-spatial feature f_i. Given a feature f_i, we can define a transactional database as follows. For each object o_i in R_i a spatial query is issued to derive a set of features $I = \{f_j : f_j \neq f_i \wedge \exists\, o_j \in R_j\ (dist(o_i, o_j) \leq \varepsilon)\}$. The collection of all feature sets I for each object in R_i defines a transactional table T_i. T_i is then mined using some itemsets mining method (e. g., [1,13]). The *frequent* feature sets I in this table, according to a minimum support value, and can be used to define rules of the form:

$$(label(o) = f_i) \Rightarrow (o \text{ close to some } o_j \in R_j, \forall f_j \in I)\,.$$

The support of a feature set I defines the confidence of the corresponding rule. For example, consider the three object-sets shown in Fig. 2. The lines indicate object pairs within a distance ε from each other. The shapes indicate different features. Assume that one must extract rules having feature a on their left-hand side. In other words, find features that occur frequently close to feature a. For each instance of a, generate an itemset; a_1 generates $\{b, c\}$ because there is at least one instance of b (e. g., b_1 and b_2) and one instance of c (e. g., c_1) close to a_1. Similarly, a_2 generates itemset $\{b\}$ (due to b_2). Let 75% be the minimum confidence. One first discovers frequent itemsets (with minimum support 75%) in $T_a = \langle \{b, c\}, \{b\} \rangle$, which gives us a sole itemset $\{b\}$. In turn, one can generate the rule

$$(label(o) = a) \Rightarrow (o \text{ close to } o_j \text{ with } label(o_j) = b)\,,$$

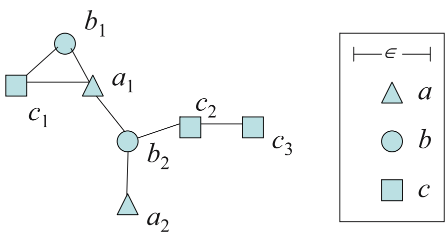
with confidence 100%. For simplicity, in the rest of the discussion, $f_i \Rightarrow I$ will be used to denote rules that associate instances of feature f_i with instances of feature sets I, $f_i \notin I$, within its proximity. For example, the rule above can be expressed by $a \Rightarrow \{b\}$. The mining process for feature a can be repeated for the other features (e. g., b and c) to discover rules having them on their left side (e. g., one can discover rule $b \Rightarrow \{a,c\}$ with conf. 100%). Note that the features on the right hand side of the rules are not required to be close to each other. For example, rule $b \Rightarrow \{a,c\}$ does not imply that for each b the nearby instances of a and c are close to each other. In Fig. 2, observe that although b_2 is close to instances a_1 and a_2 of a and instance c_2 of c, c_2 is neither close to a_1 nor to a_2.

A co-location *clique* pattern P of length k is described by a set of features $\{f_1, f_2, \ldots, f_k\}$. A valid instance of P is a set of objects $\{o_1, o_2, \ldots, o_k\}$: $(\forall\, 1 \leq i \leq k, o_i \in R_i) \wedge (\forall\, 1 \leq i < j \leq k, dist(o_i, o_j) \leq \varepsilon)$. In other words, all pairs of objects in a valid pattern instance should be close to each other, or else the closeness relationships between the objects should form a *clique graph*. Consider again Fig. 2 and the pattern $P = \{a,b,c\}$. $\{a_1, b_1, c_1\}$ is an instance of P, but $\{a_1, b_2, c_2\}$ is not.

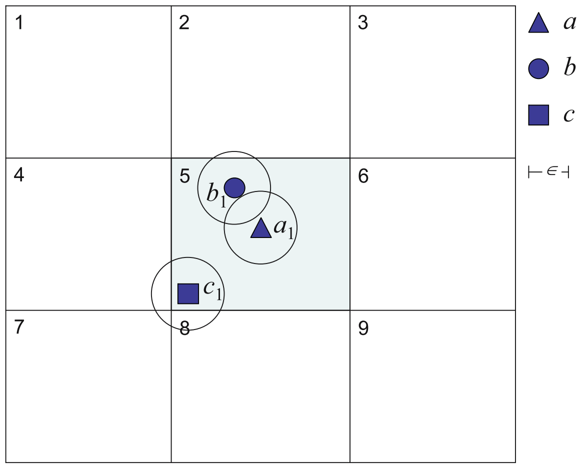
[3,10] define some useful measures that characterize the interestingness of co-location patterns. The first is the *participation ratio* $pr(f_i,P)$ of a feature f_i in pattern P, which is defined by the following equation:

$$pr(f_i, P) = \frac{\#\text{ instances of } f_i \text{ in any instance of } P}{\#\text{ instances of } f_i}\,. \tag{1}$$

Using this measure, one can define *co-location rules* that associate features with the existences of other features in their neighborhood. In other words, one can define rules of the form $(label(o) = f_i) \Rightarrow (o$ participates in an instance



Co-location Patterns, Algorithms, Figure 2 Mining example



Co-location Patterns, Algorithms, Figure 3 A regular grid and some objects

of P with confidence $pr(f_i,P)$). These rules are similar to the ones defined in [4]; the difference here is that there should be neighborhood relationships between all pairs of features on the right hand side of the rule. For example, $pr(b,\{a,b,c\}) = 0.5$ implies that 50% of the instances of b (i. e., only b_1) participate in some instance of pattern $\{a,b,c\}$ (i. e., $\{a_1, b_1, c_1\}$).

The *prevalence* $prev(P)$ of a pattern P is defined by the following equation:

$$prev(P) = \min\{pr(f_i, P), f_i \in P\} . \tag{2}$$

For example, $prev(\{b,c\}) = 2/3$ since $pr(b,\{b,c\}) = 1$ and $pr(c,\{b,c\}) = 2/3$. The prevalence captures the minimum probability that whenever an instance of some $f_i \in P$ appears on the map, it will then participate in an instance of P. Thus, it can be used to characterize the strength of the pattern in implying co-locations of features. In addition, prevalence is monotonic; if $P \subseteq P'$, then $prev(P) \geq prev(P')$. For example, since $prev(\{b, c\}) = 2/3$, we know that $prev(\{a,b,c\}) \leq 2/3$. This implies that the a priori property holds for the prevalence of patterns and algorithms like generalized [1] can be used to mine them in a level-wise manner [10].

Finally, the *confidence* $conf(P)$ of a pattern P is defined by the following equation:

$$conf(P) = \max\{pr(f_i, P), f_i \in P\} . \tag{3}$$

For example, $conf(\{b,c\}) = 1$ since $pr(b,\{b,c\}) = 1$ and $pr(c,\{b,c\}) = 2/3$. The confidence captures the ability of the pattern to derive co-location rules using the participation ratio. If P is confident with respect to a minimum confidence threshold, then it can derive at least one co-location rule (for the attribute f_i with $pr(f_i,P) = conf(P)$). In Fig. 2, $conf(\{b, c\}) = 1$ implies that we can find one feature in $\{b, c\}$ (i. e., b), every instance of which participates in an instance of $\{b, c\}$. Given a collection of spatial objects characterized by different features, a minimum prevalence threshold *min_prev*, and a minimum confidence threshold *min_conf*, a data analyst could be interested in discovering prevalent and/or confident patterns and the co-location rules derived by them. The confidence of a co-location rule between two patterns, $P_1 \rightarrow P_2$, $P_1 \cap P_2 = \varnothing$, can be defined by the conditional probability that an instance of P_1 participates in some instance of $P_1 \cup P_2$ (given that $P_1 \cup P_2$ is prevalent with respect to *min_prev*) [10].

It is now discussed how co-location patterns are mined from a spatial database. Star-like patterns are the first are of focus (as seen in Fig. 1a). As an example, consider the rule: "given a pub, there is a restaurant and a snack bar within 100 meters from it with confidence 60%". Assume that the input is n datasets $R_1, R_2, \ldots, R_n$, such that for each i, R_i stores instances of feature f_i.

The mining algorithm, a high-level description of which is shown in Fig. 4, operates in two phases; the hashing phase and the mining phase. During the hashing phase, each dataset R_i is read and the instances of the corresponding feature are spatially partitioned with the help of a regular grid. Each object is extended by the distance threshold ε to form a disk and hashed into the partitions intersected by this disk. Figure 3 shows an example. The space is partitioned into 3×3 cells. Object a_1 (which belongs to dataset R_a, corresponding to feature a) is hashed to exactly one partition (corresponding to the central cell C_5). Object b_1 is hashed to two partitions (C_2 and C_5). Finally, object c_1 is hashed into four partitions (C_4, C_5, C_7, and C_8).

The mining phase employs a main memory algorithm to efficiently find the association rules in each cell. This method is in fact a multi-way main memory spatial join algorithm based on the plane sweep technique [2,5,8]. The **synch_sweep** procedure extends the plane sweep technique used for pairwise joins to (i) apply for multiple inputs and (ii) for each instance of one input, find if there is at least one instance from other inputs close to it.

synch_sweep takes a feature f_i as input and a set of partitions of all feature instances hashed into the same cell C, and finds the maximal patterns each feature instance is included directly (without computing their sub-patterns first). The objects in the partition R_i^C (corresponding to feature f_i) in cell C are scanned in sorted order of their x-value. For each object o_i, we initialize the *maximal* star pattern L where o_i can participate as L's center. Then for each other feature, we sweep a vertical line along

```
/* R_i stores the coordinates of all objects with feature f_i */
Algorithm find_centric_co-locations(R_1, R_2, ..., R_n)
    /* 1. Spatial-hashing phase */
    super-impose a regular grid G over the map;
    for each feature f_i
        hash the objects from R_i to a number of buckets:
            each bucket corresponds to a grid cell;
            each object o is hashed to the cell(s) intersected by
                the disk centered at o with radius ϵ;
    /* 2. Mining phase */
    for each cell C of the grid;
        for each feature f_i
            load bucket R_i^C in memory;
            /* R_i^C containing objects in R_i hashed into C */
            sort points of R_i^C according to their x co-ordinate;
        for each feature f_i
            synch_sweep(f_i, R_1^C, R_2^C, ..., R_n^C);
```

Co-location Patterns, Algorithms, Figure 4
An algorithm for reference feature co-locations

the x-axis to find if there is any instance (i. e., object) within ε distance from o_i; if there is, we add the corresponding feature to L. Finally, L will contain the maximal pattern that includes f_i; for each subset of it we increase the support of the corresponding co-location rule. For more details about this process, the reader can refer to [14].

Overall, the mining algorithm requires two database scans; one for hashing and one for reading the partitions, performing the spatial joins and counting the pattern supports, provided that the powerset of all features but f_i can fit in memory. This is a realistic assumption for typical applications (with 10 or less feature types). Furthermore, it can be easily extended for arbitrary pattern graphs like those of Fig. 1b and c.

Key Applications

Sciences

Scientific data analysis can benefit from mining spatial co-location patterns [9,12]. Co-location patterns in census data may indicate features that appear frequently in spatial neighborhoods. For example, residents of high income status may live close to areas of low pollution. As another example from geographical data analysis, a co-location pattern can associate contaminated water reservoirs with a certain decease in their spatial neighborhood. Astronomers may use spatial analysis to identify features that commonly appear in the same constellation (e. g., low brightness, similar colors). Biologists may identify interesting feature combinations appearing frequently in close components of protein or chemical structures.

Decision Support

Co-location pattern analysis can also be used for decision support in marketing applications. For example, consider an E-commerce company that provides different types of services such as weather, timetabling and ticketing queries [6]. The requests for those services may be sent from different locations by (mobile or fix line) users. The company may be interested in discovering types of services that are requested by geographically neighboring users in order to provide location-sensitive recommendations to them for alternative products. For example, having known that ticketing requests are frequently asked close to timetabling requests, the company may choose to advertise the ticketing service to all customers that ask for a timetabling service.

Future Directions

Co-location patterns can be extended to include the temporal dimension. Consider for instance, a database of moving objects, such that each object is characterized by a feature class (e. g., private cars, taxis, buses, police cars, etc.). The movements of the objects (trajectories) are stored in the database as sequences of timestamped spatial locations. The objective of spatio-temporal co-location mining is to derive patterns composed by combinations of features like the ones seen in Fig. 1. In this case, each edge in the graph of a pattern corresponds to features that are close to each other (i. e., within distance ε) for a large percentage (i. e., large enough support) of their locations during their movement. An exemplary pattern is “ambulances are found close to police cars with a high probability”. Such extended spatial co-location patterns including the temporal aspect can be discovered by a direct application of the existing algorithms. Each temporal snapshot of the moving objects database can be viewed as a segment of a huge map (that includes all frames) such that no two segments are closer to each other than ε. Then, the spatio-temporal co-location patterns mining problem is converted to the spatial co-locations mining problem we have seen thus far. A more interesting (and more challenging) type of spatio-temporal collocation requires that the closeness relationship has a duration of at least τ time units, where τ is another mining parameter. For example, we may consider, as a co-location instance, a combination of feature instances (i. e., moving objects), which move closely to each other for τ *continuous* time units. To count the support of such *durable* spatio-temporal patterns, we need to slide a window of length τ along the time dimension and for each position of the window, find combinations of moving objects that qualify the pattern. Formally, given a durable pattern P, specified by a feature-relationship graph (like the ones of Fig. 1) which has a node f_i and distance/duration constraints ε and τ, the participation ratio of feature f_i in P is defined by:

$$pr(f_i, P) = \frac{\#\ \tau\text{-length windows with an instance of } P}{\#\ \tau\text{-length windows with a moving object of type } f_i} \,. \quad (4)$$

Thus, the participation ratio of a feature f_i in P is the ratio of window positions that define a sub-trajectory of at least one object of type f_i which also defines an instance of the pattern. Prevalence and confidence in this context are defined by (2) and (3), as for spatial co-location patterns. The efficient detection of such patterns from historical data as well as their on-line identification from streaming spatio-temporal data are interesting problems for future research.

Cross References

► Co-location Pattern
► Patterns, Complex
► Retrieval Algorithms, Spatial

Recommended Reading

1. Agrawal, R., Skrikant, R.: Fast algorithms for mining association rules. In: Proc. of the 20th Int. Conf. on Very Large Data Bases, pp. 487–499 (1994)
2. Brinkhoff, T., Kriegel, H.P., Seeger, B.: Efficient processing of spatial joins using r-trees. In: Proc. of ACM SIGMOD Int'l Conference (1993)
3. Huang, Y., Xiong, H., Shekhar, S., Pei, J.: Mining confident co-location rules without a support threshold. In: Proc. of the 18th ACM Symposium on Applied Computing (ACM SAC) (2003)
4. Koperski, K., Han, J.: Discovery of spatial association rules in geographic information databases. In: Proc. of the 4th Int. Symp. Advances in Spatial Databases, SSD, vol. 951, pp. 47–66 (1995)
5. Mamoulis, N., Papadias, D.: Multiway spatial joins. ACM Trans. Database Syst. **26**(4), pp. 424–475 (2001)
6. Morimoto, Y.: Mining frequent neighboring class sets in spatial databases. In Proc. of ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining. (2001)
7. Munro, R., Chawla, S., Sun, P.: Complex spatial relationships. In: Proc. of the 3rd IEEE International Conference on Data Mining (ICDM) (2003)
8. Preparata, F.P., Shamos, M.I.: Computational geometry: an introduction. Springer-Verlag New York, Inc. (1985)
9. Salmenkivi, M.: Evaluating attraction in spatial point patterns with an application in the field of cultural history. In: Proceedings of the 4th IEEE International Conference on Data Mining. (2004)
10. Shekhar, S., Huang, Y.: Discovering spatial co-location patterns: A summary of results. In: Proc of the 7th International Symposium on Advances in Spatial and Temporal Databases (SSTD). (2001)
11. Wang, J., Hsu, W., Lee, M.L.: A framework for mining topological patterns in spatio-temporal databases. In: Proceedings of the 14th ACM international conference on Information and knowledge management. Full paper in IEEE Trans. KDE, 16(12), 2004. (2005)
12. Yang, H.: Srinivasan Parthasarathy, and Sameep Mehta. Mining spatial object associations for scientific data. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (2005)
13. Zaki, M.J., Gouda, K.: Fast vertical mining using diffsets. In: Proc. of ACM SIGKDD Conference (2003)
14. Zhang, X., Mamoulis, N., Cheung, D.W.L., Shou, Y.: Fast mining of spatial collocations. In: Proc. of ACM SIGKDD Conference. (2004)

Co-location Patterns, Interestingness Measures

MARKO SALMENKIVI
HIIT Basic Research Unit, Department of Computer Science, University of Helsinki, Helsinki, Finland

Synonyms

Selection criteria; Significance measures; Association measures; Co-location patterns; Interestingness measures

Definition

Interestingness measures for spatial *co-location patterns* are needed to select from the set of all possible patterns those that are in some (quantitatively measurable) way, characteristic for the data under investigation, and, thus, possibly, provide useful information.

Ultimately, interestingness is a subjective matter, and it depends on the user's interests, the application area, and the final goal of the spatial data analysis. However, there are properties that can be objectively defined, such that they can often be assumed as desirable. Typically, these properties are based on the frequencies of pattern instances in the data.

Spatial association rules, co-location patterns and co-location rules were introduced to address the problem of finding associations in spatial data, and in a more general level, they are applications of the problem of finding *frequent patterns* on spatial domain. Interestingness of a pattern in data is often related to its frequency, and that is the reason for the name of the problem.

In practice, a pattern is considered as interesting, if the values of the interestingness measures (possibly only one) of the pattern exceed the thresholds given by the user.

Historical Background

Finding patterns in data and evaluating their interestingness has traditionally been an essential task in statistics. Statistical data analysis methods cannot always be applied to large data masses. For more detailed discussion of the problems, see Scientific Fundamentals. Data mining, or knowledge discovery from databases, is a branch of computer science that arose in the late 1980s, when classical statistical methods could no longer meet the requirements of analysis of the enormously increasing amount of digital data. Data mining develops methods for finding trends, regularities, or patterns in very large datasets. One of the first significant contributions of data mining research was the notion of association rule, and algorithms, e. g., Apriori [1], for finding all interesting association rules from transaction databases. Those algorithms were based on first solving the subproblem of the *frequent itemset discovery*. The interesting association rules could easily be deduced from the frequent itemsets.

When applying association rules in spatial domain, the key problem is that there is no natural notion of transactions, due to the continuous two-dimensional space. Spatial association rules were first introduced in [11]. They were analogous to association rules with the exception that at least one of the predicates in a spatial association rule expresses spatial relationship (e. g., *adjacent_to, within, close_to*). The rules always continue a reference feature. Support and confidence were used as interestingness measures similarly to the transaction-based association rule mining. Another transaction-based approach was proposed in [9]: spatial objects were grouped into disjoint partitions. One of the drawbacks of the method is that different partitions may result in different sets of transactions, and, thus, different values for the interestingness measures of the patterns. As a solution to the problem, co-location patterns in the context of the event-centric model were introduced in [10].

Scientific Fundamentals

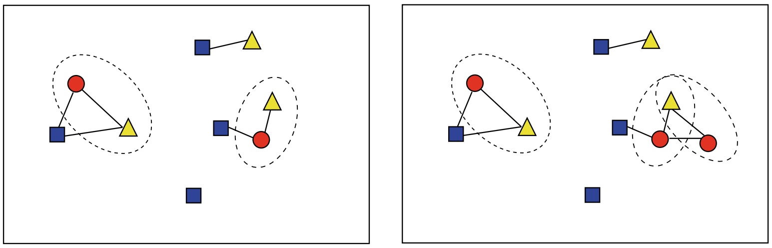
Different models can be employed to model the spatial dimension, and the interpretation of co-location patterns as well as the interestingness measures are related to the selected model. The set of proposed models include at least the window-centric model, reference feature-centric model, event-centric model, and buffer-based model [14].

Co-location patterns and co-location rules can be considered in the general framework of *frequent pattern* mining as pattern classes. Other examples of pattern classes are itemsets and association rules (in relational databases), episodes (in event sequences), strings, trees and graphs. [7,15]

In the window-centric model the space is discretized by a uniform grid, and the set of all the possible windows of size $k \times k$ form the set of transactions. The items of the transaction are the features present in the corresponding window. Thus, support can be used as the interestingness measure. The interpretation of the confidence of the rule $A \rightarrow B$ is the conditional probability of observing an instance of B in an arbitrary $k \times k$-window, given that an instance of feature A occurs in the window.

The reference feature-centric model focuses on a specific *Boolean spatial feature*, and all the discovered patterns express relationships of the reference feature and other features. The spatial association rules introduced in [11] are based on selecting a reference feature, and then creating transactions over space. Transactions make it possible to employ the interestingness measures introduced for transaction databases in the context of *frequent itemset discovery*: *support* of a feature set (analogously to the support of an itemset in transaction databases), and *confidence* (or *conditional probability*) of an association rule.

In the event-centric model introduced in [10], the spatial proximity of objects is modeled by using the notion of *neighborhood*. The neighborhood relation $R(x, y)$, x, $y \in O$, where O is the set of spatial objects, is assumed to be given as input. The objects and the neighborhood relation can be represented as an undirected graph, where nodes correspond to objects, and an edge between nodes indicates that the objects are neighbors (see Fig. 1). A limitation of the event-centric model is that it can be used only when the objects are points. An advantage is that the pattern discovery is not restricted to patterns with a reference feature. Furthermore, no explicit transactions need to be formed. This fact also has consequences as to the choice of relevant interestingness measures. In a transaction-based model a single object can only take part in one transaction,



Co-location Patterns, Interestingness Measures, Figure 1 Examples of (row) instances of co-location patterns in the event-centric model

whereas in the event-centric model it is often the case that a single object participates in several instances of a particular pattern.

Figure 1 shows an example. There are nine spatial point objects. The set of features consists of three features indicated by a triangle (denote it by A), circle (B), and rectangle (C). In this example only one feature is assigned to each object, in general there may be several of them. There are three instances of feature A, two instances of B, and four instances of C. The solid lines connect the objects that are neighbors. Cliques of the graph indicate the instances of co-location patterns. Hence, there is only one instance of pattern $\{ABC\}$ containing all the features.

The participation ratio of a feature f in a co-location pattern $\mathcal{P}$ is the number of instances of the feature that participate in an instance of $\mathcal{P}$ divided by the number of all instances of f. For instance, in the example data on the left panel of Fig. 1 the participation ratio of feature A in pattern $\{AB\}$, $pr(A,\{AB\}) = 2/3$, since two out of three instances of feature A also participate in instances of $\{AB\}$. Correspondingly $pr(B,\{AB\}) = 2/2 = 1$, since there is no instance of B that is not participating in $\{AB\}$. The objects on the right panel of Fig. 1 are equal to those of the left panel, except for an additional point with feature B. Now, there are two different points with feature B such that they both are neighbors of the same instance of A. The instances of pattern $\{A, B\}$ have been indicated by the dashed lines. Thus, one instance of A participates in two instances of $\{A, B\}$. The participation ratios are equal to the left-side case: $pr(A, \{AB\}) = 2/3$ and $pr(B, \{AB\}) = 3/3 = 1$.

Prevalence of a co-location pattern is defined as $prev(P) = \min\{pr(f, \mathcal{P}), f \in \mathcal{P}\}$. A co-location pattern is *prevalent*, if its prevalence exceeds the user-specified threshold value. Prevalence is a monotonous interestingness measure with respect to the pattern class of co-location patterns, since adding features to P can clearly only decrease $prev(P)$.

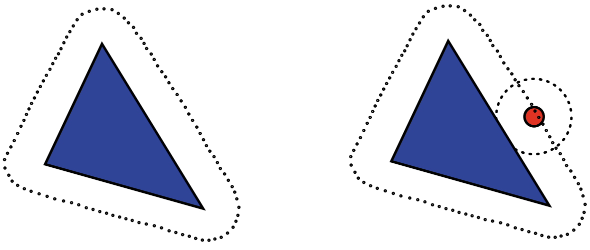
Let $\mathcal{P}$ and $\mathcal{Q}$ be co-location patterns, and $\mathcal{P} \cap \mathcal{Q} = \emptyset$. Then $\mathcal{P} \rightarrow \mathcal{Q}$ is a *co-location rule*. The *confidence* (or *conditional probability*) of $\mathcal{P} \rightarrow \mathcal{Q}$ (in a given dataset) is the fraction of instances of $\mathcal{P}$ such that they are also instances of $\mathcal{P} \cup \mathcal{Q}$. A co-location rule is *confident* if the confidence of the rule exceeds the user-specified threshold value. A "sufficiently" high prevalence of a co-location pattern indicates that the pattern can be used to generate confident co-location rules. Namely, assume that the user-specified confidence threshold for interesting co-location rules is *min_conf*. Then, if $prev(P) \geq min\_conf$, rule $f \rightarrow f_1, \ldots, f_n$ is confident for all $f \in \mathcal{P}$.

In the example of Fig. 1 the prevalence $prev(AB) = \min(2/3,1) = 2/3$. Thus, one can generate rules $A \rightarrow B$, the confidence of rule being 2/3, and $B \rightarrow A$ (confidence 1).

Another interestingness measure proposed for co-location patterns is *maximum participation ratio* (MPR). Prevalence of a pattern is the minimum of the participation rations of its features, whereas MPR is defined as the *maximum* of them. Correspondingly, a "sufficiently" high MPR implies that at least one of the features, denote it by T, rarely occurs outside $\mathcal{P}$. Hence, the co-location rule $\{T\} \rightarrow \mathcal{P} \setminus \{T\}$ is confident [4]. The motivation of using the MPR is that rare features can more easily be included in the set of interesting patterns.

A drawback of MPR is that it is not monotonous. However, a weaker property ("weak monotonicity") can be proved for MPR. This property is utilized in [4] to develop a level-wise search algorithm for mining confident co-location rules.

The buffer-based model extends the co-location patterns to polygons and line strings [14]. The basic idea is to introduce a buffer, which is a zone of a specified distance, around each spatial object. The boundary of the buffer is the isoline of equal distance to the edge of the objects (see Fig. 2). The (Euclidean) neighborhood $\mathcal{N}(o)$ of an object o



Co-location Patterns, Interestingness Measures, Figure 2 Examples of neighborhoods in the buffer-based model is the area covered by its buffer. The (Euclidean) neighborhood of a feature f is the union of $\mathcal{N}(o_i)$, where $o_i \in O_f$, and O_f is the set of instances of f. Further, the (Euclidean) neighborhood $\mathcal{N}(C)$ for a feature set $C = \{f_1, f_2, \ldots, f_n\}$ is defined as the intersection of $\mathcal{N}(f_i), f_i \in C$.

The coverage ratio $Pr(C)$, where $C = \{f_1, f_2, \ldots, f_n\}$ is a feature set is defined as $\frac{\mathcal{N}(C)}{Z}$, where Z is the total size of the investigation area. Intuitively, the coverage ratio of a set of features measures the fraction of the investigation area that is influenced by the instances of the features.

The coverage ratio is a monotonous interestingness measure in the pattern class of co-location patterns in the buffer-based model, with respect to the size of the co-location pattern [14]. Now in the buffer-based model the conditional probability (confidence) of a co-location rule $P \to Q$ expresses the probability of finding the neighborhood of Q in the neighborhood of P. Due to the monotonicity of coverage ratio, it can be computed as $\frac{\mathcal{N}(P \cup Q)}{\mathcal{N}(P)}$. Xiong et al. also demonstrate that the definition of conditional probability (confidence) of a co-location rule in the event-centric model does not satisfy the law of compound probability: it is possible that $Prob(BC|A) \neq Prob(C|AB)$ $Prob(B|A)$, where $Prob(BC|A)$ is equal to the confidence of the rule $A \to BC$. They show, however, that in the buffer-based model this law holds.

Statistical Approaches

An essential difference in the viewpoints of *spatial statistics* and co-location pattern mining is that in statistics the dataset is considered as a sample. The aim in statistics is typically to infer, based on the sample, knowledge of properties of the "reality", that is, the phenomenon, that generated the data. The goal of co-location pattern mining is to find descriptions of the data, that is, only the content of the available database is the object of investigation. In a sense, statistical analysis is more ambitious. However, sophisticated statistical data analysis methods cannot always be applied to large data masses. This may be due to the lack of computational resources, expert knowledge, or other human resources needed to preprocess the data before statistical analysis is possible.

Furthermore, depending on the application, treating the content of a spatial database as a sample may be relevant, or not. Consider, for instance, roads represented in a spatial database. Clearly, it is usually the case that (practically) all of them are included in the database, not only a sample. On the other hand, in an ecological database that includes the known locations of nests of different bird species, it is obvious that not all the nests have been observed, and thus a part of the information is missing from the database. Another example is a linguistic database that contains dialect variants of words in different regions. Such variants cannot in practice be exhaustively recorded everywhere, and, thus, the data in the database is a sample.

Statistical analysis of spatial point patterns is closely related to the problem of finding interesting co-location patterns (see, e. g., [2,3]). In statistics, features are called *event types*, and their instances are *events*. The set of events in the investigation area form a *spatial point pattern*. Point patterns of several event types (called *marked point patterns*) may be studied, for instance, to evaluate spatial correlation (either positive, i. e., clustering of events, or negative, i. e., repulsion of events). Analogously, the point pattern of a single event type can be studied for evaluating possible *spatial autocorrelation*, that is, clustering or repulsion of the events of the event type.

In order to evaluate spatial (auto)correlation, point patterns, that is the data, are modeled as realizations (samples) generated by spatial point processes. A spatial point process defines a joint probability distribution over all point patterns. The most common measures of spatial correlation in point patterns are the $G(h)$, and $K(h)$-functions. For a single event type the value of $G(h)$-function in data is the number of events such that the closest other event is within a distance less than h divided by the number of all events. For two event types, instead of the closest event of the same type, the closest event of the *other* event type is considered. Thus, the confidence of the co-location rule $A \to B$, where A and B are single features in the event-centric model, is equal to the value of $G_{A,B}(h)$-function in the data, when the neighborhood relation is defined as the maximum distance of h between objects.

The statistical framework implies that the relationship of the phenomenon and the data, which is a sample, has to be modeled in some way. In spatial statistics, the interestingness measures can be viewed from several perspectives, depending on the statistical framework, and the methods used in the data analysis. One of the most common frameworks is the hypothesis testing.

Hypothesis testing sets up a null hypothesis, typically assuming no correlation between features, and an alternative hypothesis that assumes spatial correlation. A test statistic, e. g., $G(h)$ or $K(h)$-function, for measuring spatial correlation is selected, denote it by T. The value of the test statistic in data, denote it by t, is compared against the theoretical distribution of the test statistic, assuming that the null hypothesis holds. Then, a natural interestingness measure of the observed spatial correlation is based on the so-called p-value, which is defined as $Pr(T > t | H_0)$. The smaller the p-value, the smaller the probability that the observed degree of spatial correlation could have been occurred by chance. Thus, the correlation can be interpreted as interesting if the p-value is small. If the p-value is not greater than a predefined α, the deviation is defined to be *statistically significant* with the *significance level* α.

The correlation patterns introduced in [12] represent an intermediate approach between spatial point pattern analysis and co-location pattern mining. Correlation patterns are defined as interesting co-location patterns (in the event-centric model) of the form $A \rightarrow B$, where A and B are single features. The interestingness is determined by the statistical significance of the deviation of the observed $G(h)$-value from a null hypothesis assuming no spatial correlation between features A and B.

Key Applications

Large spatial databases and spatial datasets. Examples: digital road map [13], census data [6], place name data [5,12].

Future Directions

A collection of interesting patterns can be regarded as a summary of the data. However, the pattern collections may be very large. Thus, condensation of the pattern collections and pattern ordering are important challenges for research on spatial co-location patterns.

Co-location patterns and rules are local in the sense that, given a pattern, only the instances of the features that appear in the pattern are taken into account when evaluating the interestingness of the pattern. However, the overall distribution and density of spatial objects and features may, in practice, provide significant information as to the interestingness of a pattern. This challenge is to some extent related to the challenge of integrating statistical and data mining approaches.

Cross References

- ► Co-location Pattern
- ► Co-location Pattern Discovery
- ► Data Analysis, Spatial
- ► Frequent Itemset Discovery
- ► Frequent Pattern
- ► Statistical Descriptions of Spatial Patterns

Recommended Reading

1. Agrawal, R.; Ramakrishnan, S.: Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases, pp. 487–499, September 12–15 (1994)
2. Bailey, T.C., Gatrell, A.C.: Interactive Spatial Data Analysis. Longman, Harlow (1995)
3. Diggle, P.J.: Statistical Analysis of Spatial Point Patterns. Mathematics in Biology. Academic, London (1983)
4. Huang, Y., Xiong, H., Shekhar, S., Pei, J.: Mining confident co-location rules without a support threshold. Proc. 2003 ACM Symposium on Applied Computing (ACM SAC March, 2003), pp. 497–501, Melbourne, Florida (2003)
5. Leino, A., Mannila, H., Pitkänen, R.: Rule discovery and probabilistic modeling for onomastic data. In: Lavrac, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) Knowledge Discovery in Databases: PKDD 2003. Lecture Notes in Artificial Intelligence 2838, pp. 291–302. Springer, Heidelberg (2003)
6. Malerba, D., Esposito, F., Lisi, F.A.: Mining spatial association rules in census data. In: Proceedings of 4th International Seminar on New Techniques and Technologies for Statistics (NTTS 2001), Crete (2001)
7. Mannila, H., Toivonen, H., Verkamo, A.I.: Discovering frequent episodes in sequences. In: First International Conference on Knowledge Discovery and Data Mining (KDD'95, August), pp. 210–215, Montreal, Canada, AAAI Press (1995)
8. Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Mining Knowl Disc **1**(3), 241–258 (1997)
9. Morimoto, Y.: Mining frequent neighboring class sets in spatial databases. In: Int Proc. 7th ACM SIGKDD Conference on Knowledge and Discovery and Data Mining, pp. 353–358, San Francisco, California (2001)
10. Shekhar, S., Huang, Y.: Discovering spatial co-location patterns: a summary of results. In: Proceedings of 7th International Symposium on Advances in Spatial and Temporal Databases (SSTD 2001), Redondo Beach, California (2001)
11. Koperski, K., Han, J.: Discovery of spatial association rules in geographic information databases. In: Proc. of 4th International Symposium on Large Spatial Databases (SSD95), Portlane, Maine, pp. 47–66 (1995)
12. Salmenkivi, M.: Efficient mining of correlation patterns in spatial point data. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) Knowledge Discovery in Databases: PKDD-06, Berlin, Germany, Proceedings. Lecture Notes in Computer Science 4213, pp. 359–370. Springer, Berlin (2006)
13. Shekhar, S., Ma, X.: GIS subsystem for a new approach to accessing road user charges
14. Xiong, H., Shekhar, S., Huang, Y., Kumar, V., Ma, X., Yoo, J.S.: A framework for discovering co-location patterns in data sets with extended spatial objects. In: Proceedings of the Fourth SIAM International Conference on Data Mining (SDM04), Lake Buena Vista, Florida, (2004)
15. Zaki, M.J.: Efficiently mining frequent trees in a forest. In: Proc. of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2002)

Co-location Rule Discovery

► Co-location Pattern Discovery

Co-location Rule Finding

► Co-location Pattern Discovery

C

Co-location Rule Mining

► Co-location Pattern Discovery

Collocation, Spatio-temporal

► Movement Patterns in Spatio-temporal Data

COM/OLE

► Smallworld Software Suite

Combinatorial Map

► Geosensor Networks, Qualitative Monitoring of Dynamic Fields

Components

► Smallworld Software Suite

Composite Geographic Information Systems Web Application

► GIS Mashups

Computational Grid

► Grid

Computational Infrastructure

► Grid

Computer Cartography

► Conflation of Geospatial Data

Computer Environments for GIS and CAD

JOE ASTROTH
Autodesk Location Services, San Rafael, CA, USA

Synonyms

CAD and GIS platforms; Evolution of GIS and LBS; Technological inflection points in GIS and CAD development; Convergence of GIS and CAD; Geo-mashups

Definition

In the last few decades, computing environments have evolved to accommodate the need for integrating the separate, and often incompatible, processes of Geographic Information Systems (GIS) and Computer Assisted Design (CAD). This chapter will explore the evolution of GIS and CAD computing environments—from desktop to Web, and finally to wireless—along with the industry requirements that prompted these changes.

Historical Background

Before the 1980s, Computer Assisted Design (CAD) and Geographic Information Systems (GIS) functions were performed primarily on minicomputers running 32-bit operating systems such as VAX, VMS, or UNIX. Since minicomputers were expensive (approximately $200,000), many CAD and GIS solutions were bundled with hardware and offered as turnkey solutions. Although the combination of software and hardware was a popular option, it was still prohibitively expensive for small organizations, making CAD and GIS affordable only for government, academic institutions, and major corporations.

With the advent of the personal computer, particularly the IBM PC in 1981, which sold for approximately $1600, GIS and CAD became affordable for small- and medium-sized organizations.

Soon after the introduction of affordable personal computers, Autodesk developed the first PC-based CAD software, AutoCAD®, which sold for approximately $1000. Desktop GIS products appeared on the market shortly thereafter. Rather than being retrofitted from minicomputer programs, the most successful of these applications were engineered specifically for the PC. With the availability of these powerful desktop programs, small- to medium-sized organizations that had previously relied on analog mapping and drafting had access to the wealth of information and time-saving tools formerly available only to large organizations.

Scientific Fundamentals

CAD and GIS During the Workstation Phase

Although both GIS and CAD were introduced on the PC at around the same time, they were considered completely separate, and often incompatible, applications, making data sharing difficult. For example, precision and accuracy in GIS, unlike that of CAD, is variable, depending on scale. In CAD, precision was represented as 64-bit units (double precision), and in GIS as 32-bit units (single precision). Positional accuracy, which indicates the proximity of a feature on a map to its real location on earth, is quite high for CAD relative to GIS. For example, a 1:10,000 scale map might have a positional accuracy of 2.5m (8.2 feet).

Another barrier to data sharing between CAD and GIS on the PC was the process of data collection. Many GIS survey instruments, such as Total Stations and GPS, collect data in ground units, rather than grid units. Ground units, which represent features on the earth's surface exactly, are both longer and bigger than grid units. In addition, elevations and scale are not factored into ground units.

Since a great deal of map data was collected in the field, maps drawn in CAD were stored in ground units, and scale and coordinate systems were added afterwards. CAD engineers found that using ground units, rather than grid units, was advantageous. For example, assuming that dimensions on the map were accurate, if the line on the map measured 100 meters (328 feet), it corresponded to 100 meters on the ground. With GIS grid units, 100 meters on the map might actually correspond to 100.1 meters (328 feet 3 inches) on the ground.

Another significant difference between CAD and GIS applications is that in CAD, unlike in GIS, points and polylines represent objects in the real world, but contain no attached information. In GIS, points, polylines, and polygons can represent wells, roads, and parcels, and include attached tables of information. In many cases, when spatial information was transferred from CAD to GIS applications, features were "unintelligent" and had to be assigned meaningful information, such as topology and attributes, manually. Because of this manual step, translation from CAD to GIS was extremely difficult, even with automated import tools.

GIS and CAD applications also provided different types of tools, making it difficult for users to switch systems. In GIS applications, tools were designed for data cleanup, spatial analysis, and map production, whereas tools in CAD were intended for data entry, drafting, and design. Since CAD drafting tools were much easier to use, CAD technicians were wary of using GIS software for creation of design drawings.

CAD drawings themselves also made it difficult to transfer data to GIS. A typical CAD drawing contains objects made up of arcs and arbitrarily placed label text. However, in GIS, text can be generated based on attributes or database values, often producing a result that is not aesthetically pleasing to a cartographer.

The representation of land parcels, common in GIS applications for municipalities, presented another challenge for integrating CAD drawings and GIS. Polylines portraying lot and block lines in a survey plan need to be translated into meaningful polygons in GIS that represent the parcel. Cleanup tools are used to ensure the accuracy of the lot and block lines. Each parcel must also be associated with its appropriate attributes, and a polygon topology must be created so that Parcel Identification Numbers (PINs or PIDs) inside each polygon are linked to the parcel database.

These barriers to integrating GIS and CAD led to the development of software solutions in each phase of the technological advancement in computing environments.

C

Bridging the Gap Between CAD in GIS in the Workstation Phase In the initial workstation phase, the only way to integrate GIS data with AutoCAD data was to use DXF™ (drawing exchange format). This process was extremely time-consuming and error-prone. Many CAD drawings were drawn for a particular project or plan and never used again. Often these drawings were not in the same coordinate system as the GIS and had to be transformed on import. Even today, a GIS enterprise is built and maintained by importing data from CAD drawings. Graphic representations of layers of a formation, such as water, sewer, roads and parcels, are imported into the GIS using the file-based method.

To better merge the CAD world with the GIS world, a partnership was formed between Autodesk, Inc. and ESRI, leading to the creation of ArcCAD®. ArcCAD was built on AutoCAD and enabled users to create GIS layers and to convert GIS layers into CAD objects. This tool also facilitated data cleanup and the attachment of attributes. Because ArcCAD enabled GIS data to be shared with a greater number of people, the data itself became more valuable.

Although ArcCAD solved some of the integration problems between CAD and GIS, it still did not provide full GIS or CAD functionality. For example, overlay analysis still had to be performed in ArcInfo® and arcs and splines were not available in the themes created by ArcCAD.

In order to provide a fully functional GIS built on the AutoCAD platform, Autodesk developed AutoCAD Map® (now called Autodesk Map®), which made it simple for a CAD designer to integrate with external databases, build topology, perform spatial analysis, and utilize data cleaning, without file translation or lost data. In AutoCAD Map, lines and polygons were topologically intelligent with regard to abstract properties such as contiguity and adjacency. Since DWG™ files were already file-based packets of information, they became GIS-friendly when assigned topology and connected to databases. Precision was enforced instantly, since the DWG files could now store coordinate systems and perform projections and transformations. AutoCAD Map represented the first time a holistic CAD and GIS product was available for the PC Workstation environment.

Although AutoCAD Map could import and export the standard GIS file types (circa 1995: ESRI SHP, ESRI Coverage, ESRI E00, Microstation DGN, MapInfo MID/MIF, Atlas BNA) users began to request real-time editing of layers from third-party GIS files. To meet this demand, Autodesk created a new desktop GIS/CAD product called Autodesk World®. World was designed for users who were not GIS professionals or AutoCAD engineers, and offered the basic tools of both systems: precision drafting and the capability to query large geospatial data and perform rudimentary analysis and reports.

World used a Microsoft Office interface to access and integrate different data types, including geographic, database, raster, spreadsheet, and images, and supported Autodesk DWG as a native file format, increasing the value of maps created in AutoCAD and AutoCAD Map. World enabled users to open disparate GIS data files simultaneously and perform analysis regardless of file type. Autodesk World could access, analyze, edit and save data in all the standard formats without import or export.

Although Autodesk World represented a real breakthrough in integrating GIS and CAD files, it lacked an extensive CAD design environment. AutoCAD was still the CAD environment of choice, and AutoCAD Map continued to offer better integration of GIS within a full CAD environment. Autodesk World filled a need, much like other desktop GIS solutions at the time, but there was still a gap between the CAD design process and analysis and mapping within the GIS environment.

In the same time period, AutoCAD Map continued to evolve its GIS capabilities for directly connecting, analyzing, displaying, and theming existing GIS data (in SDE, SHP, DGN, DEM, and Raster formats, for example) without import or export. In support of the Open GIS data standard, AutoCAD Map could read OpenGIS information natively. GIS and CAD integration continues to be one of key features of AutoCAD Map.

CAD and GIS During the Web Phase

The next significant inflection point in technology was the World Wide Web, which increased the number of users of spatial data by an order of magnitude. With the advent of this new technology and communication environment, more people had access to information than ever before.

Initially, CAD and GIS software vendors responded to the development of the Web by Web-enabling existing PC applications. These Web-enabled applications offered the ability to assign Universal Resource Locators (URLs) to graphic objects or geographic features, such as points, lines and polygons, and enabled users to publish their content for viewing in a browser as an HTML (Hypertext Markup Language) page and a series of images representing maps or design.

Software developers also Web-enabled CAD and GIS software by providing a thin client or browser plug-in, which offered rich functionality similar to the original application.

CAD for the Web In the early Web era, slow data transfer rates required thin clients and plug-ins to be small (less than one megabyte) and powerful enough to provide tools such as pan and zoom. In light of this, Autodesk's developed a CAD plug-in called Whip! which was based on AutoCAD's ADI video driver.

Although the Whip! viewer today has evolved into the Autodesk DWF™ Viewer, the file format, DWF (Design Web Format) remains the same. DWF files can be created with any AutoCAD based product, including AutoCAD Map, and the DWF format displays the map or design on the Web as it appears on paper. DWF files are usually much smaller than the original DWGs, speeding their transfer across the Web. With the development of DWF, Internet users had access to terabytes of information previously available only in DWG format. This was a milestone in information access.

From a GIS perspective, 2D DWF files were useful strictly for design and did not represent true coordinate systems or offer GIS functionality. Although Whip!-based DWF was extremely effective for publishing digital versions of maps and designs, GIS required a more comprehensive solution. Note: Today, DWF is a 3D format that supports coordinate systems and object attributes.

GIS for the Web As the Web era progressed, it became clear that a simple retrofit of existing applications would not be sufficient for Web-enabled GIS. In 1996, Autodesk purchased MapGuide® from Argus Technologies. MapGuide viewer was a browser plug-in that could display full vector-format GIS data streamed from an enormous repository using very little bandwidth. Each layer in MapGuide viewer could render streamed data from different MapGuide Servers around the Internet. For example, road layers could be streamed directly from a server in Washington, DC, while the real-time location of cars could be streamed directly from a server in Dallas, Texas. MapGuide managed its performance primarily with scale-dependent authoring techniques that limited the amount of data based on the current scale of the client map.

MapGuide could perform basic GIS functions such as buffer and selection analysis, as well as address-matching navigation with zoom-goto. One of the more powerful aspects of MapGuide was the generic reporting functionality, in which MapGuide could send a series of unique IDs of selected objects to any generic Web page for reporting. Parcels, for example, could be selected in the viewer and the Parcel IDs could be sent to a server at City Hall that had the assessment values. A report was returned, as a Web page, containing all the information about the selected parcels. Again, the report could reside on any server, anywhere. The maps in MapGuide were just stylized pointers to all the potential servers around the Internet, containing spatial and attribute data. MapGuide was revolutionary at the time, and represented, in the true sense, applications taking advantage of the distributed network called the Web.

MapGuide continued to evolve, using ActiveX controls for Microsoft Internet Explorer, a plug-in for Netscape and a Java applet that could run on any Java-enabled browser. Initially, MapGuide used only its own file format, SDF, for geographic features. Later, MapGuide could natively support DWG, DWF, SHP, Oracle Spatial, and ArcSDE.

Although MapGuide was an extremely effective solution, it could run only on Microsoft Windows servers. The development of MapGuide OpenSource and Autodesk MapGuide Enterprise was inspired by the need to move toward a neutral server architecture and plug-in-free client experience. MapGuide could be now be used either without a plug-in or with the newest DWF Viewer as a thin client.

Within AutoCAD Map, users could now publish directly to the MapGuide Server and maintain the data dynamically, further closing the GIS-CAD gap.

CAD and GIS During the Wireless Phase

Wireless CAD and GIS marked the beginning of the next inflection point on the information technology curve, presenting a new challenge for GIS and CAD integration. Since early wireless Internet connection speeds were quite slow—approximately one quarter of wired LAN speed—Autodesk initially decided that the best method for delivering data to handheld device was “sync and go,” which required physically connecting a handheld to a PC and using synchronization software to transfer map and attribute data to the device. GIS consumers could view this data on their mobile devices in the field without being connected to a server or desktop computer. Since handheld devices were much less expensive than PCs, mobile CAD and GIS further increased the number of people who had access to geospatial information.

Wireless CAD Autodesk OnSite View (circa 2000) allowed users to transfer a DWG file to Palm-OS handheld and view it on the device. When synchronized, the DWG file was converted to an OnSite Design file (OSD), and when viewed, allowed users to pan, zoom and select features on the screen.

With the advent of Windows CE support, OnSite View allowed redlining, enabling users to mark up a design without modifying the original. Redlines were saved as XML (Extensible Markup Language) files on the handheld and were transferred to the PC on the next synchronization or docking. These redline files could be imported into AutoCAD, where modifications to the design could be made.

Autodesk OnSite View could be considered more mobile than wireless, since no direct access to the data was available without connecting the mobile device to the PC. OnSite View filled a temporary niche before broadband wireless connections became available.

Wireless GIS and Location-Based Services Initially, the mobile GIS solution at Autodesk was OnSite Enterprise, which leveraged the mobility of OnSite and the dynamism of MapGuide. OnSite Enterprise created handheld MapGuide maps in the form of OSD files that users could simply copy off the network and view on their mobile devices with OnSite.

In 2001, when true broadband wireless came on the horizon, Autodesk created a new corporate division focused solely on Location-Based Services (LBS). The burgeoning Wireless Web required a new type of software, designed specifically to meet the high transaction volume, performance (+ 40 transactions per second), and privacy requirements of wireless network operators (WNOs). The next technological inflection point had arrived, where maps and location-based services were developed for mass-market mobile phones and handheld devices.

Autodesk Location Services created LocationLogic™, a middleware platform that provides infrastructure, application services, content provisioning, and integration services for deploying and maintaining location-based services. The LocationLogic platform was built by the same strong technical leadership and experienced subject matter experts that worked on the first Autodesk GIS products. The initial version of LocationLogic was a core Geoserver specifically targeted for wireless and telecom operators that required scalability and high-volume transaction throughput without performance degradation.

The LocationLogic Geoserver was able to provide:

- Point of Interest (POI) queries
- Geocoding and reverse geocoding
- Route planning
- Maps
- Integrated user profile and location triggers

Points of Interest (POIs) usually comprise a set of businesses that are arranged in different categories. POI directories, which can include hundreds of categories, are similar to Telecom Yellow Pages, but with added location intelligence. Common POI categories include Gas Stations, Hotels, Restaurants, and ATMs, and can be customized for each customer. Each listing in the POI tables is spatially indexed so users can search for relevant information based on a given area or the mobile user's current location.

Geocoding refers to the representation of a feature's location or address in coordinates (x,y) so that it can be indexed spatially, enabling proximity and POI searches within a given area. Reverse geocoding converts x, y coordinates to a valid street address. This capability allows the address of a mobile user to be displayed once their phone has been located via GPS or cell tower triangulation. Applications such as "Where am I?" and friend or family finders utilize reverse geocoding.

Route planning finds the best route between two or more geographical locations. Users can specify route preferences, such as shortest path based on distance, fastest path based on speed limits, and routes that avoid highways, bridges, tollways, and so on. Other attributes of route planning include modes of transportation (such as walking, subway, car), which are useful for European and Asian countries.

The maps produced by the LocationLogic's Geoserver are actually authored in Autodesk MapGuide. Although the Geoserver was built "from the ground up," LocationLogic was able to take advantage of MapGuide's effective mapping software.

LocationLogic also supports user profiles for storing favorite routes or POIs. Early versions of LocationLogic also allowed applications to trigger notifications if the mobile user came close to a restaurant or any other point of interest. This capability is now used for location-based advertising, child zone notifications, and so on.

Key Applications

Early LBS applications built on LocationLogic included traffic alerts and friend finder utilities. For example, Verizon Wireless subscribers could receive TXT alerts about traffic conditions at certain times of day and on their preferred routes. Friend finder utilities alerted the phone user that people on their list of friends were within a certain distance of the phone.

More recently, Autodesk Location Services has offered two applications built on LocationLogic that can be accessed on the cell phone and via a Web browser: Autodesk Insight™ and Autodesk Family Minder.

Autodesk Insight is a service that enables any business with a PC and Web browser to track and manage field workers who carry mobile phones. Unlike traditional fleet tracking services, Insight requires no special investment in GPS hardware. Managers and dispatchers can view the locations of their staff, determine the resource closest to a customer site or job request, and generate turn-by-turn travel instructions from the Web interface. Managers can also receive alerts when a worker arrives at a given location or enters or leaves a particular zone. Reports on travel, route histories, and communications for groups or individuals, over the last 12 or more months, can be generated from the Web interface.

Family Minder allows parents and guardians to view the real-time location of family members from a Web interface or their handset. Parents and guardians can also receive notifications indicating that a family member has arrived at or left a location. The recent advances in mobile phone technology, such as sharper displays, increased battery life and strong processing power, make it possible for users to view attractive map displays on regular handsets.

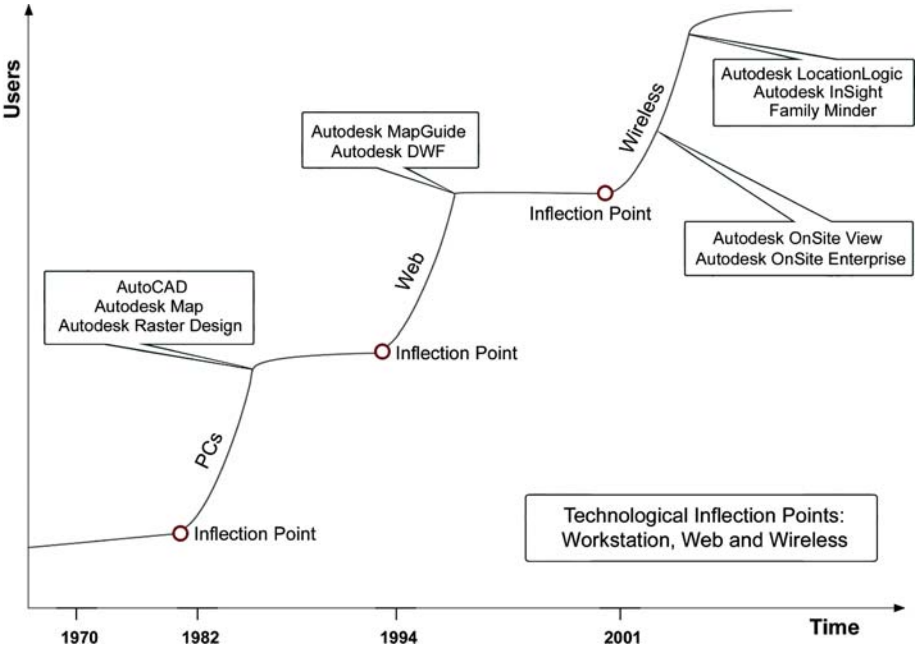
Enterprise GIS: Workstation, Web and Wireless Synergy

In 1999, Autodesk acquired VISION*®, along with its expertise in Oracle and enterprise GIS integration. This was a turning point for Autodesk GIS. File-based storage of information (such as DWG) was replaced with enterprise database storage of spatial data. Currently, Autodesk has integrated VISION* into its development, as seen in Autodesk GIS Design server. Autodesk Topobase™, which also stores its data in Oracle, connects to AutoCAD Map and MapGuide to provide enterprise GIS Public Works and Municipal solutions.

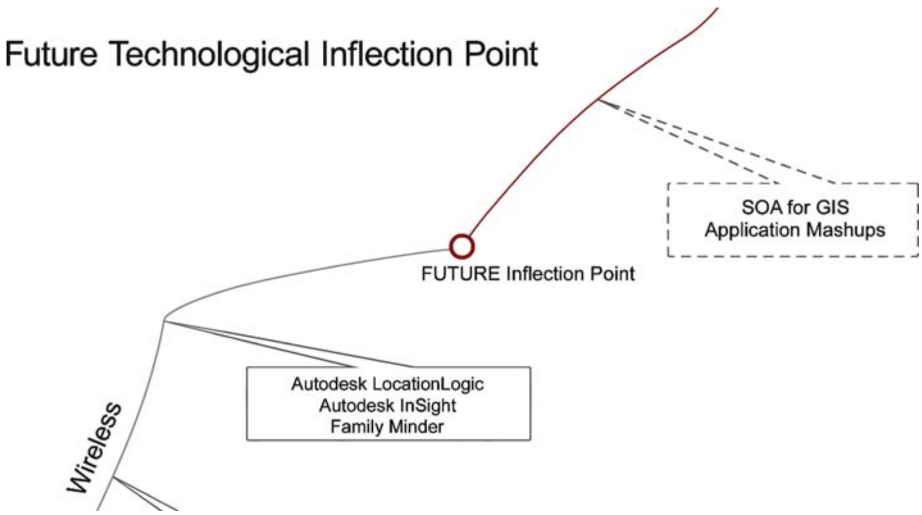
MapGuide and AutoCAD Map support Oracle Spatial and Locator, which allow all spatial data to be stored in a central repository. All applications can view the data without duplication and reliance on file conversion. AutoCAD Map users can query as-built information from the central repository for help in designs, and any modifications are saved and passed to GIS users. The central GIS database can also be published and modified from Web-based interfaces, such as MapGuide. Real-time wireless applications, such as Autodesk Insight, can use the repository for routing and mobile resource management.

Summary

At each technological inflection point—workstation, Web and wireless—Autodesk has leveraged infrastructural changes to exponentially increase the universe of potential consumers of geospatial information. The shift from minicomputer to PC saw Autodesk create AutoCAD and AutoCAD Map to enable sharing of geographic and design information. The next inflection point, workstation to Web, spurred another jump in the number of spatial data consumers, and the CAD and GIS gap continued to close. The most recent inflection point, Web to wireless, saw the num-



Computer Environments for GIS and CAD, Figure 1 Technological Inflection Points along the Information Technology Curve – Exponential Jumps in Access to Geospatial Information



Computer Environments for GIS and CAD, Figure 2 Future Technological Inflection Point (continued from Fig. 1)

ber of spatial data users reach a new high, as GIS applications were embedded in the users' daily tools, such as cell phones (see Fig. 1). At this point in the technology curve, the need for synergy between CAD and GIS is apparent more than ever. Since the value of spatial data increases exponentially with the number of users who have access to it, Autodesk's enterprise GIS solution, with its centralized spatial database, provides significant value to a wide variety of spatial data consumers.

Autodesk has a history of leveraging inflection points along the computing and communication technology curve to create exciting and innovative solutions. For over two decades, Autodesk's mission has been to spearhead the "democratization of technology" by dramatically increasing the accessibility of heretofore complex and expensive software. This philosophy has been pervasive in the GIS and LBS solutions that it has brought to a rapidly growing geospatial user community.

Future Directions

The next potential inflection point will emerge with the development of Service Oriented Architecture (SOA), built upon a Web 2.0 and Telco 2.0 framework. Not only will the distributed data and application architecture continue to increase the number of geospatial data consumers, but it will increase the use and accessibility of powerful analytical and visual tools as well.

Historically, the Web was leveraged to distribute data with wireless and Web technology. Now, the "geo-mashups" between tools such as Google Earth and AutoCAD Civil 3D, make use of the interaction of Web-based applications and data. A simple example of an SOA application is LocationLogic's geocoder, which performs geocoding and reverse-geocoding via Asynchronous JavaScript and XML (AJAX) calls to a URL that return sets of coordinates and addresses, respectively.

As GIS applications become integrated into current technologies (such as cars and courier boxes), demand for rapid data and application processing will apply pressure to all aspects of the distribution model. One challenge will be to provide rapid information updates, such as current aerial photographs and the latest traffic conditions. These "just-in-time" applications will require a massive increase in scale to accommodate the large number of business and personal users. At each technological inflection point, the accessibility to this vital information will increase exponentially (see Fig. 2).

CAD and GIS will soon be so integrated that the location on the timeline from design to physical feature or survey to map will be the only way to determine which technology is currently being used. Seamless and transparent services and data distribution will bring subsets of CAD and GIS utilities together to produce dynamic applications on demand. Servers will no longer host only databases, but will run self-supported applications, functions, and methods that are CAD, GIS, database, and business oriented. These services will be offered through the new Web 2.0 to provide powerful solutions.

Transparent GIS Services and integrated geospatial data will affect a larger segment of the population. No longer will the technology just be "cool," but will be completely integral to daily life. Autodesk's role will be to continue to provide tools that will leverage this new reality and meet the coming new demands in information and technology.

Cross References

► Data Models in Commercial GIS Systems
► Information Services, Geography
► Internet GIS
► Internet-Based Spatial Information Retrieval
► Location Intelligence
► Location-Based Services: Practices and Products
► Oracle Spatial, Raster Data
► Privacy Threats in Location-Based Services
► Vector Data
► Web Mapping and Web Cartography
► Web Services, Geospatial

Recommended Reading

1. Autodesk Geospatial (http://images.autodesk.com/adsk/files/autodesk_geospatial_white_paper.pdf)
2. CAD and GIS – Critical Tools, Critical Links: Removing Obstacles Between CAD and GIS Professionals (http://images.autodesk.com/adsk/files/3582317_CriticalTools0.pdf)
3. Best Practices for Managing Geospatial Data (http://images.autodesk.com/adsk/files/%7B574931BD-8C29-4A18-B77C-A60691A06A11%7D_Best_Practices.pdf)
4. Autodesk, Inc.: Map 3D 2007 Essentials. San Rafael, Autodesk Press, (2007)
5. Vance, D., Walsh, D., Eisenberg, R.: Inside AutoCAD Map 2000. Autodesk Press, San Rafael, (2000)
6. Sharma, C.: Wireless Internet Enterprise Applications. Wiley Computer Publishing, New York, NY., Wiley Tech Brief Series (2001)
7. Barry, D.: Web Services and Service-Orientated Architectures, The Savvy Manager's Guide, Your Road Map to Emerging IT. Morgan Kaufmann Publishers, San Francisco, CA. (2003)
8. Hjelm, J.: Creating Location Services for the Wireless Guide. Wiley Computer Publishing, New York, Professional Developer's Guide Series (2002)
9. Schiller, J., Voisard, A.: Location-Based Services. Morgan Kaufmann Publishers, San Francisco, CA. (2004)
10. Kolodziej, K., Hjelm, J.: Local Positioning Systems, LBS Applications and Services. CRC Press, Boca Raton, FL., Taylor and Francis Group (2006)
11. Laurini, R., Thompson, D.: Fundamentals of Spatial Information Systems. Academic Press, London & San Diego, CA, The APIC Series (1992)
12. Jagoe, A.: Mobile Location Services, The Definitive Guide. Prentice Hall Publisher, Upper Saddle River, NJ. (2003)
13. Plewe, B.: GIS ONLINE; Information retrieval, Mapping and the Internet. OnWord Press, Santa Fe, NM. (1997)
14. Vance, D., Smith, C., Appell, J.: Inside Autodesk World. OnWord Press, Santa Fe, NM. (1998)
15. Vance, D., Eisenberg, R., Walsh, D.: Inside AutoCAD Map 2000, The Ultimate How-To Resource and Desktop Reference for AutoCAD Map. OnWord Press, Florence (2000)
16. Longley, P., Goodchild, M., Maguire, D., Rhind, D.: Second Edition, Geographical Information Systems, Volume 1 Principles and Technical Issues, Volume 2 Management Issues and Applications. John Wiley & Sons Publishing, New York, NY. (1999)

Computer Supported Cooperative Work

► Geocollaboration

Computing Fitness of Use of Geospatial Datasets

LEEN-KIAT SOH, ASHOK SAMAL
Department of Computer Science and Engineering,
University of Nebraska, Lincoln, NE, USA

Synonyms

Quality of information; Dempster shafer belief theory; Conflict resolution; Information fusion; Timeseries data; Frame of discernment; Plausibility; Evidence

Definition

Geospatial datasets are widely used in many applications including critical decision support systems. The goodness of the dataset, called the Fitness of Use (FoU), is used in the analysis and has direct bearing on the quality of derived information from the dataset that ultimately plays a role in decision making for a specific application. When a decision is made based on different sources of datasets, it is important to be able to fuse information from datasets of different degrees of FoU. Dempster-Shafer belief theory is used as the underlying conflict resolution mechanism during information fusion. Furthermore, the Dempster-Shafer belief theory is demonstrated as a viable approach to fuse information derived from different approaches in order to compute the FoU of a dataset.

Historical Background

In most applications, sometimes it is assumed that the datasets are perfect and without any blemish. This assumption is, of course, not true. The data is merely a representation of a continuous reality both in space and time. It is difficult to measure the values of a continuous space and time variable with infinite precision. Limitations are also the result of inadequate human capacity, sensor capabilities and budgetary constraints. Therefore, the discrepancy exists between the reality and the datasets that are derived to represent it. It is especially critical to capture the degree of this discrepancy when decisions are made based on the information derived from the data. Thus, this measure of quality of a dataset is a function of the purpose for which it is used, hence it is called its *fitness of use* (FoU). For a given application, this value varies among the datasets. Information derived from high-FoU datasets is more useful and accurate for the users of the application than that from low-FoU datasets. The challenge is to develop appropriate methods to fuse derived information of varying degrees of FoU as well as of derived information from datasets of varying degrees of FoU. This will give insights as to how the dataset can be used or how appropriate the dataset is for a particular application [1].

An information theoretic approach is used to compute the FoU of a dataset. The Dempster-Shafer belief theory [2] is used as the basis for this approach in which the FoU is represented as a range of possibilities and integrated into one value based on the information from multiple sources. There are several advantages of the Dempster-Shafer belief theory. First, it does not require that the individual elements follow a certain probability. In other words, Bayes theory considers an event to be either true or untrue, whereas the Dempster-Shafer allows for unknown states [4]. This characteristic makes the Dempster-Shafer belief theory a powerful tool for the evaluation of risk and reliability in many real applications when it is impossible to obtain precise measurements and results from real experiments. In addition, the Dempster-Shafer belief Theory provides a framework to combine the evidence from multiple sources and does not assume disjoint outcomes [3]. Additionally, the Dempster-Shafer's measures are not less accurate than Bayesian methods, and in fact reports have shown that it can sometimes outperform Bayes' theory [5,6].

C

Scientific Fundamentals

Assume that there is a set of geospatial datasets, $S = \{S_1, S_2, \cdots, S_n\}$. A dataset S_i may consist of many types of information including (and not limited to) spatial coordinates, metadata about the dataset, denoted by aux_i, and the actual time series data, denoted by ts_i.

The metadata for a dataset may include the type of information being recorded (e. g., precipitation or volume of water in a stream), the period of record, and the frequency of measurement. Thus,

$$aux_i = \langle type_i, tb_i, te_i, int_i \rangle ,$$

where tb_i and te_i denote the beginning and the ending time stamps for the measurements, and int_i is the interval at which the measurements are made. Other metadata such as the type and age of the recording device can also be added.

The time series data in a dataset may consist of a sequence of measurements,

$$ts_i = \left\langle m_{i,1}, m_{i,2}, \ldots, m_{i,p} \right\rangle .$$

Each measurement stores both the time the measurement was taken and the actual value recorded by the sensor. Thus, each measurement is given by

$$m_{i,j} = \left\langle t_{i,j}, v_{i,j} \right\rangle .$$

It is assumed that the measurements in the dataset are kept in chronological order. Therefore,

$$t_{i,j} < t_{i,k} , \quad \text{for} \quad j < k .$$

Furthermore, the first and last measurement times should match the period of record stored in the metadata,

$$tb_i = t_{i,1} \quad \text{and} \quad te_i = t_{i,p} .$$

The problem of finding the suitability of a dataset for a given application is to define a function for the FoU that computes the fitness of use of a dataset described above. The function *FoU* maps S_i to a normalized value between 0 and 1:

$$FoU\,(S_i, A) = [0, 1]\,,$$

where S_i is a single dataset and A is the intended application of the data. The application A is represented in the form of domain knowledge that describes how the goodness of a dataset is viewed. A set of rules may be used to specify this information. Thus,

$$A = \{R_1, R_2, \cdots, R_d\}\ ,$$

where R_i is a domain rule that describes the goodness of a dataset and d is the number of rules. Therefore, the *FoU* function is defined with respect to an application domain. Different applications can use different rules for goodness and derive different FoU values for the same dataset.

Dempster-Shafer Belief Theory

The two central ideas of the Dempster-Shafer belief theory are: (a) obtaining degrees of belief from subjective probabilities for a related question, and (b) Dempster's rule for combining such degrees of belief when they are based on independent items of evidence. For a given proposition P, and given some evidence, a confidence interval is derived from an interval of probabilities within which the true probability lies within a certain confidence. This interval is defined by the *belief* and *plausibility* supported by the evidence for the given proposition. The lower bound of the interval is called the *belief* and measures the strength of the evidence in favor of a proposition. The upper bound of the interval is called the *plausibility*. It brings together the evidence that is compatible with the proposition and is *not inconsistent* with it. The values of both belief and plausibility range from 0 to 1. The belief function (*bel*) and the plausibility function (*pl*) are related by:

$$pl(P) = 1 - bel(\overline{P})\,,$$

where $\overline{P}$ is the negation of the proposition P. Thus, $bel(\overline{P})$ is the extent to which evidence is in favor of $\overline{P}$.

The term *Frame of Discernment* (FOD) consists of all hypotheses for which the information sources can provide evidence. This set is finite and consists of mutually exclusive propositions that span the hypotheses space. For a finite set of mutually exclusive propositions (θ), the set of possible hypotheses is its power set (2^θ), i. e., the set of all possible subsets including itself and a null set. Each of these subsets is called a focal element and is assigned a confidence interval [belief, plausibility].

Based on the evidence, a probability mass is first assigned to each focal element. The masses are *probability-like* in that they are in the range [0, 1] and sum to 1 over all hypotheses. However, they represent the belief assigned to a focal element. In most cases, this basic probability assignment is derived from the experience and the rules provided by some experts in the application domain.

Given a hypothesis H, its belief is computed as the sum of all the probability masses of the subsets of H as follows:

$$bel(H) = \sum_{e \subset H} m(e)\,,$$

where $m(e)$ is the probability mass assigned to the subset e. The probability mass function distributes the values on subsets of the frame of discernment. Only to those hypotheses, for which it has direct evidence, are the none-zero values assigned. Therefore, the Dempster-Shafer belief theory allows for having a single piece of evidence supporting a set of multiple propositions being true. If there are multiple sources of information, probability mass functions can be derived for each data source. These mass values are then combined using Dempster's Combination Rule to derive joint evidence in order to support a hypothesis from multiple sources. Given two basic probability assignments, m_A and m_B for two independent sources (A and B) of evidence in the same frame of discernment, the joint probability mass, m_{AB}, can be computed according to Dempster's Combination Rule:

$$m_{AB}(C) = \frac{\sum\limits_{A \cap B = C} m(A) * m(B)}{1 - \sum\limits_{A \cap B = \emptyset} m(A) * m(B)}\,.$$

Furthermore, the rule can be repeatedly applied for more than two sources sequentially, and the results are order-independent. That is, combining different pieces of evidence in different sequences yields the same results.

Finally, to determine the confidence in a hypothesis H being true, belief and plausibility are multiplied together:

$$\mathit{confidence}(H) = bel(H) \cdot pl(H)\,.$$

Thus, the system is highly confident regarding a hypothesis being true if it has high belief and plausibility for that hypothesis being true.

Suppose that there are three discrete FoU outcomes of the datasets *suitable* (s), *marginal* (m), and *unsuitable* (u), and $\theta = \{s,m,u\}$. Then, the frame of discernment is

$$\begin{aligned} FOD &= 2^\theta \\ &= \{\emptyset, \{s\}, \{m\}, \{u\}, \{s, m\}, \{s, u\}, \{m, u\}, \{s, m, u\}\}\,. \end{aligned}$$

To illustrate how the Dempster-Shafer belief theory can be used to fuse derived information from geospatial databases, two novel approaches to derive information from

C

geospatial databases are herein presented – i. e., (1) heuristics, and (2) statistics – before fusing them. For each approach, the computation of the FoU of the dataset based on the derived information is demonstrated.

In the first approach, a set of domain heuristics for this purpose and then the combination rule to compute the FoU of the datasets are used. The heuristics can be based on common sense knowledge or can be based on expert feedback. The following criteria are used:

- *Consistency* – A dataset is consistent if it does not have any gaps. A consistent dataset has a higher fitness value
- *Length* – The period of record for the dataset is also an important factor in the quality. Longer periods of record generally imply a higher fitness value
- *Recency* – Datasets that record more recent observations are considered to be of a higher fitness value
- *Temporal Resolution* – Data are recorded at different time scales (sampling periods). For example, the datasets can be recorded daily, weekly or monthly. Depending on the application, higher or lower resolution may be better. This is also called the granularity [7]
- *Completeness* – A data record may have many attributes, e. g., time, location, and one or more measurements. A dataset is complete if all the relevant attributes are recorded. Incomplete datasets are considered to be inferior [7]
- *Noise* – All datasets have some noise due to many different factors. All these factors may lead to data not being as good for use in applications.

For each of the above criteria, one or more heuristics can be defined to determine the probability mass for different data quality values. The heuristics in the form of rules are specified as follows:

$$C_1(S_i) \wedge C_2(S_i) \wedge \cdots \wedge C_n(S_i) \rightarrow mass(S_i, \{qtype\}) = m,$$

where C_i specifies a condition of the dataset, $C_j(S_i)$ evaluates to true if the condition C_j holds for the dataset S_i, and $mass(S_i, \{qtype\})$ denotes the mass of evidence that the dataset S_i contributes to the FoU outcome types in $\{qtype\}$. Then, a rule is triggered or fires if all the conditions are met. When the rule fires, the right-hand side of the rule is evaluated which assigns a value m to the probability mass for a given set of outcome types which in the example, $\{qtype\} \subseteq \{suitable, marginal, unsuitable\}$.

Applying a set of rules as defined above to dataset S_i thus yields a set of masses for different combinations of outcome types. These masses are then combined using the Dempster's Combination Rule to yield a coherent set of masses for each element of FOD. The result may be further reduced by considering only the singletons: {*suitable*},{*marginal*}, and {*unsuitable*}, which allows one to compute the belief, plausibility, and confidence values on only these three outcome types.

Now, how one may compute the FoU of a dataset using information derived statistically is shown. In the following, the statistical method used is called temporal variability analysis. Suppose that S_i has the following measurements:

$$ts_i = \langle m_{i,1}, m_{i,2}, \ldots, m_{i,p} \rangle, \text{ and } m_{i,j} = \langle t_{i,j}, v_{i,j} \rangle .$$

Suppose that the measurements are collected periodically at some regular intervals. Suppose that the *period* of a data series is defined as the time between two measurements collected at the same spatial location at the same time mark. Given this notion of periodicity, the average value of all measurements at each particular time mark over the interval of measurements can be computed. Formally,

$$ts_i = \langle m_{i,1}, m_{i,2}, \cdots, m_{i,p} \rangle \quad \text{can be re-written as:}$$

$$ts_i = \langle m_{i,1}, m_{i,2}, \ldots, m_{i,period}, m_{i,period+1}, m_{i,period+2}, \cdots, m_{i,2*period}, m_{i,2*period+1}, \cdots, m_{i,k*period} \rangle ,$$

such that $t_{i,k*period} - t_{i,1} = \text{int}_i$. Given the above representation, the periodic mean can be derived at each time mark j as

$$mean_{i,j} = \frac{\sum_{p=0}^{k} m_{i,p*period+j}}{k} .$$

Likewise, the periodic variance can be derived for the time marks j as

$$var_{i,j} = \frac{k \sum_{p=0}^{k} m^2_{i,p*period+j} - \left(\sum_{p=0}^{k} m_{i,p*period+j} \right)^2}{k(k-1)} .$$

Given the set of means and variances for all time marks in a period, the coefficient of variation at each time mark j can be further computed as

$$cov_{i,j} = \frac{\sqrt{var_{i,j}}}{mean_{i,j}} .$$

The temporal variability of the dataset S_i can then be defined as the average value of coefficient of variation for all time marks:

$$\bar{c}(S_i) = \frac{\sum_{j=1}^{period} cov_{i,j}}{period} .$$

Heuristics can then be used to assign probability masses to the different outcomes based on the value of $\bar{c}$. For example, to assign probability masses to the outcomes, the temporal variability can be divided into three ranges: the upper (largest) one-third, the middle one-third and the lower (smallest) one-third. For each range, one or more heuristics are defined to determine the probability mass for different FoU values. The heuristics are specified in the form of rules as

$$(\bar{c}\,(S_i)\ \ within\ range\ k) \rightarrow mass\,(S_i, \{qtype\}) = m\,,$$

where $\overline{c}(S_i)$ is the average coefficient of variation of the dataset S_i, and the range k is one the three ranges mentioned above. For a given dataset S_i, the right hand side of the above rule is evaluated and a value m to the probability mass is assigned for a given type (*suitable*, *marginal*, or *unsuitable*). These probability masses can also be combined using Dempster's Combination Rule.

Thus, at this point, there are two pieces of derived FoU values for a dataset. One is through the heuristic approach and the other through the statistical approach. Both FoU values represent a confidence in the dataset belonging to a particular type (suitable, marginal, or unsuitable). To obtain one single composite FoU out of the two values, yet another fusion can be performed. That is, to fuse the two derived information of varying FoU, one may simply treat each FoU as a mass for the dataset to belong to a particular *qtype*. Thus, by employing Dempster's Combination Rule, one can repeat the same process in order to obtain different mass values that support the notion that the dataset has a FoU of a certain type. This allows the FoUs to be fused at subsequent levels as well.

The idea of FoU has also been applied in several different contexts. De Bruin et al. [8] have proposed an approach based on decision analysis where value of information and value of control were used. Value of information is the expected desirability of reducing or eliminating uncertainty in a chance node of a decision tree while value of control is the expected amount of control that one could affect the outcome of an uncertain event. Both of these values can be computed from the probability density of the spatial data. Vasseur et al. [9] have proposed an ontology-driven approach to determine fitness of use of datasets. The approach includes conceptualization of the question and hypothesis of work to create the ontology of the problem, browsing and selecting existing sources informed by the metadata of the datasets available, appraisal of the extent of these databases matching or missing the expected data with the quality expected by the user, translation of the corresponding (matched) part into a query on the database, reformulation of the initial concepts by expanding the ontology of the problem, querying the actual databases with the query formulae, and final evaluation by the user to accept or reject the retrieved results. Further, Ahonen-Rainio and Kraak [10] have investigated the use of sample maps to supplement the fitness for use of geospatial datasets.

Key Applications

The key category of applications for the proposed technique is to categorize or cluster spatio-temporal entities (objects) into similar groups. Through clustering, the techniques can be used to group a dataset into different clusters for knowledge discovery and data mining, classification, filtering, pattern recognition, decision support, knowledge engineering and visualization.

- *Drought Mitigation:* Many critical decisions are made based on examining the relationship between the current value of a decision variable (e. g., precipitation) and its historic norms. Incorporating the fitness of use in the computation will make the decision making process more accurate.
- *Natural Resource Management:* Many natural resources are currently being monitored using a distributed sensor network. The number of these networks continues to grow as the sensors and networking technologies become more affordable. The datasets are stored as typical time series data. When these datasets are used in various applications, it would be useful to incorporate the fitness of use values in the analysis.

Future Directions

The current and future work focuses on extending the above approach to compute the fitness of use for derived information and knowledge. FoU, for example, is applied directly to raw data. However, as data is manipulated, fused, integrated, filtered, cleaned, and so on, the derived metadata or information appears, and with it, an associated measure of fitness. This fitness can be based on the intrinsic FoU of the data that the information is based on and also on the *technique* that derives the information. For example, one may say that the statistical approach is more rigorous than the heuristic approach and thus should be given more mass or confidence. Likewise, this notion can be extended to knowledge that is the result of using information. A piece of knowledge is, for example, a decision. Thus, by propagating FoU from data to information, and from information to knowledge, one can tag a decision with a confidence value.

Cross References

► Crime Mapping and Analysis
► Data Collection, Reliable Real-Time

▶ Error Propagation in Spatial Prediction
▶ Indexing and Mining Time Series Data

Recommended Reading

1. Yao, X.: Research Issues in Spatio-temporal Data Mining. A White Paper submitted to the University Consortium for Geographic Information Science (UCGIS) workshop on Geospatial Visualization and Knowledge Discovery, Lansdowne, Virginia, 18–20 Nov 2003
2. Shafer, G.: A Mathematical Theory of Evidence. Princeton Univ. Press, Princeton (1976)
3. Sentz K, Ferson S (2002) Combination of Evidence in Dempster-Shafer belief theory. Available via SANDIA Technical Report SAND2002–0835. http://www.sandia.gov/epistemic/Reports/SAND2002-0835.pdf
4. Konks D, and Challa S (2005) An Introduction to Bayesian and Dempster-Shafer Data Fusion Available via DSTO-TR-1436, Edinburgh, Australia, November 2005. http://www.dsto.defence.gov.au/publications/2563/DSTO-TR-1436.pdf
5. Cremer, F., den Breejen, E., Schutte, K.: Sensor Data Fusion for Antipersonnel Land Mine Detection. In: Proceedings of EuroFusion98, pp. 55–60. Great Malvern, UK (1998)
6. Braun, J.: Dempster-Shafer Theory and Bayesian Reasoning in Multisensor Data FuSion. Sensor Fusion: Architectures, Algorithms and Applications IV; Proceedings of SPIE 4051, pp. 255–266. (2000)
7. Mihaila, G., Raschid, L., Vidal, M.E.: Querying, "Quality of Data" Metadata. In: Proceedings of the Third IEEE Meta-Data Conference, Bethesda, Maryland, April 1999
8. De Bruin, S., Bregt, A., Van De Ven, M.: Assessing Fitness for Use: The Expected Value of Spatial Data Sets. International Journal of Geographical Information Sciences **15**(5), 457–471 (2001)
9. Vasseur, B., Devillers, R., Jeansoulin, R. : Ontological Approach of the Fitness of Use of Geospatial Datasets. In: Proceedings of 6th AGILE Conference on Geographic Information Science, pp. 497–504. Lyon, France (2003)
10. Ahonen-Rainio, P., Kraak M.J.: Deciding on Fitness for Use: Evaluating the Utility of Sample Maps as an element of Geospatial Metadata. Cartography and Geographic Information Science **32**(2), 101–112 (2005)

Computing Performance

▶ Network GIS Performance

Conceptual Generalization of Databases

▶ Abstraction of GeoDatabases

Conceptual Model

▶ Application Schema

Conceptual Modeling

▶ Spatio-temporal Database Modeling with an Extended Entity-Relationship Model

Conceptual Modeling of Geospatial Databases

▶ Modeling with ISO 191xx Standards

Conceptual Neighborhood

Anthony G. Cohn
School of Computing, University of Leeds, Leeds, UK

Synonyms

Continuity network; Qualitative similarity; Closest topological distance

Definition

A standard assumption concerning reasoning about spatial entities over time is that change is continuous. In qualitative spatial calculi, such as the mereotopological RCC or 9-intersection calculi in which a small finite set of *jointly exhaustive and pairwise disjoint* sets of relations are defined, this can be represented as a *conceptual neighborhood* diagram (also known as a *continuity network*). A pair of relations R_1 and R_2 are conceptual neighbors if it is possible for R_1 to hold at a certain time, and R_2 to hold later, with no third relation holding in between. The diagram to be found in the definitional entry for *mereotopology* illustrates the conceptual neighborhood for RCC-8.

Cross References

▶ Knowledge Representation, Spatial
▶ Mereotopology
▶ Representing Regions with Indeterminate Boundaries

Recommended Reading

1. Cohn, A.G., Hazarika, S.M.: Qualitative Spatial Representation and Reasoning: An Overview. Fundam. Inf. **46**(1–2), 1–29 (2001)
2. Cohn, A.G., Renz, J.: Qualitative Spatial Representation and Reasoning. In: Lifschitz, V., van Harmelen, F., Porter, F. (eds.) Handbook of Knowledge Representation, Ch. 13. Elsevier, München (2007)

Conceptual Schema

▶ Application Schema

Concurrency Control for Spatial Access

JING DAI, CHANG-TIEN LU
Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA

Synonyms

Concurrent spatial operations; Simultaneous spatial operations; Concurrency control protocols

Definition

Concurrency control for spatial access method refers to the techniques providing the serializable operations in multi-user spatial databases. Specifically, the concurrent operations on spatial data should be safely executed and follow the ACID rules (i. e., Atomicity, Consistency, Isolation, and Durability). With concurrency control, multi-user spatial databases can process the search and update operations correctly without interfering with each other.
The concurrency control techniques for spatial databases have to be integrated with particular spatial access methods to process simultaneous operations. There are two major concerns in concurrency control for the spatial access method. One is how to elevate the throughput of concurrent spatial operations and the other is concerned with preventing phantom access.

Main Text

In the last several decades, spatial data access methods have been proposed and developed to manage multi-dimensional databases as required in GIS, computer-aided design, and scientific modeling and analysis applications. In order to apply the widely studied spatial access methods in real applications, particular concurrency control protocols are required for multi-user environments. The simultaneous operations on spatial databases need to be treated as exclusive operations without interfering with one another.
The existing concurrency control protocols mainly focus on the R-tree family. Most of them were developed based on the concurrency protocols on the B-tree family. Based on the locking strategy, these protocols can be classified into two categories, namely, link-based approaches and lock-coupling methods.
Concurrency control for spatial access methods is generally required in commercial database management systems. In addition, concurrency control methods are required in many specific spatial applications, such as the taxi management systems that need to continuously query the locations of taxies. The study on spatial concurrency control is far behind the research on spatial query processing approaches. There are two interesting and emergent directions in this field. One is to apply concurrency control methods on complex spatial operations, such as nearest neighbor search and spatial join; the other to design concurrency control protocols for moving object applications.

Cross References

► Indexing, Hilbert R-tree, Spatial Indexing, Multimedia Indexing

Recommended Reading

1. Chakrabarti, K., Mehrotra, S.: Efficient Concurrency Control in Multi-dimensional Access Methods. In: Proceedings of ACM SIGMOD International Conference on Management of Data, Philadelphia, Pennsylvania, 1–3 June 1999
2. Kornacker, M., Mohan, C., Hellerstein,J.: Concurrency and Recovery in Generalized Search Trees. In: Proceedings of ACM SIGMOD International Conference on Management of Data, Tucson, Arizona, 13–15 May 1997
3. Song, S.I., Kim, Y.H., Yoo, J.S.: An Enhanced Concurrency Control Scheme for Multidimensional Index Structure. IEEE Transactions on Knowledge and Data Engineering, vol. 16, No. 1, pp. 97–111 (January, 2004)

Concurrency Control for Spatial Access Method

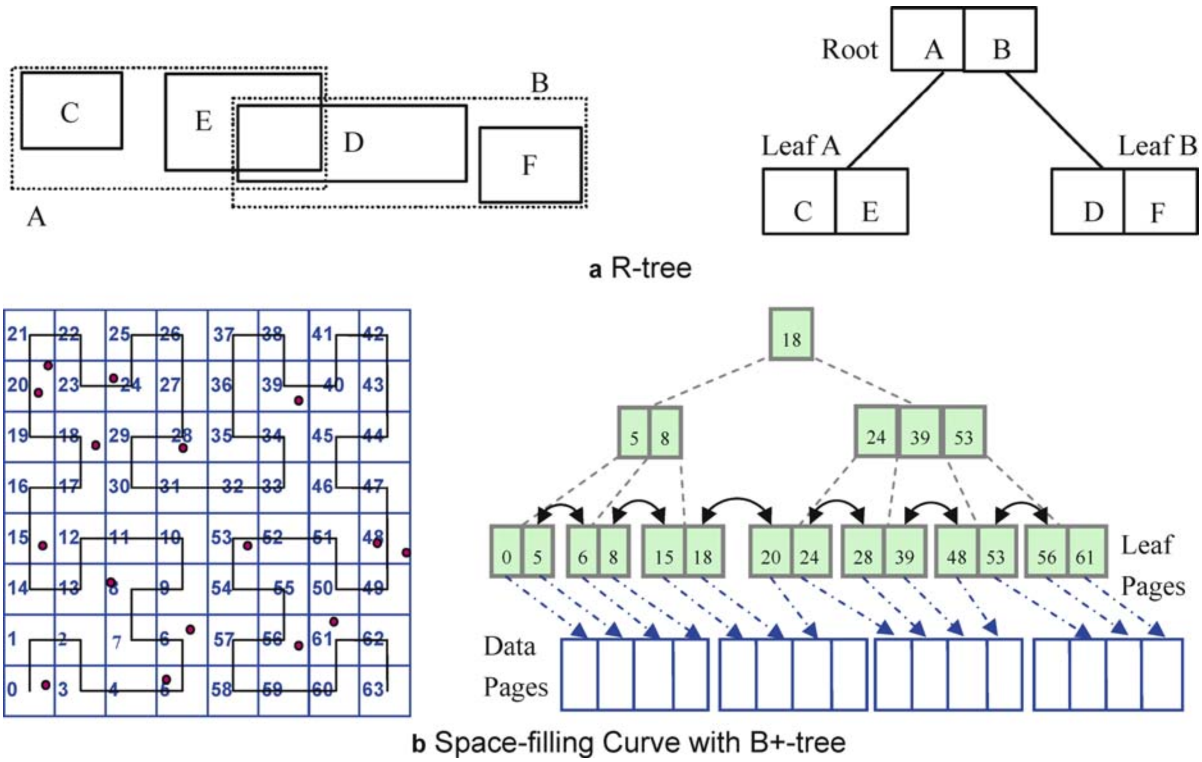
JING DAI, CHANG-TIEN LU
Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA

Synonyms

Concurrent spatial operations; Simultaneous spatial operations; Concurrency control protocols; Phantom Update Protection

Definition

The concurrency control for spatial access method refers to the techniques providing the serializable operations in multi-user spatial databases. Specifically, the concurrent operations on spatial data should be safely executed and follow the ACID rules (i. e., Atomicity, Consistency, Isolation, and Durability). With concurrency control, multi-user spatial databases can perform the search and update operations correctly without interfering with each other.
There are two major concerns in the concurrency control for spatial data access. One is the throughput of concurrent spatial operations. The throughput refers to the number of



Concurrency Control for Spatial Access Method, Figure 1 Representative Spatial Access Methods

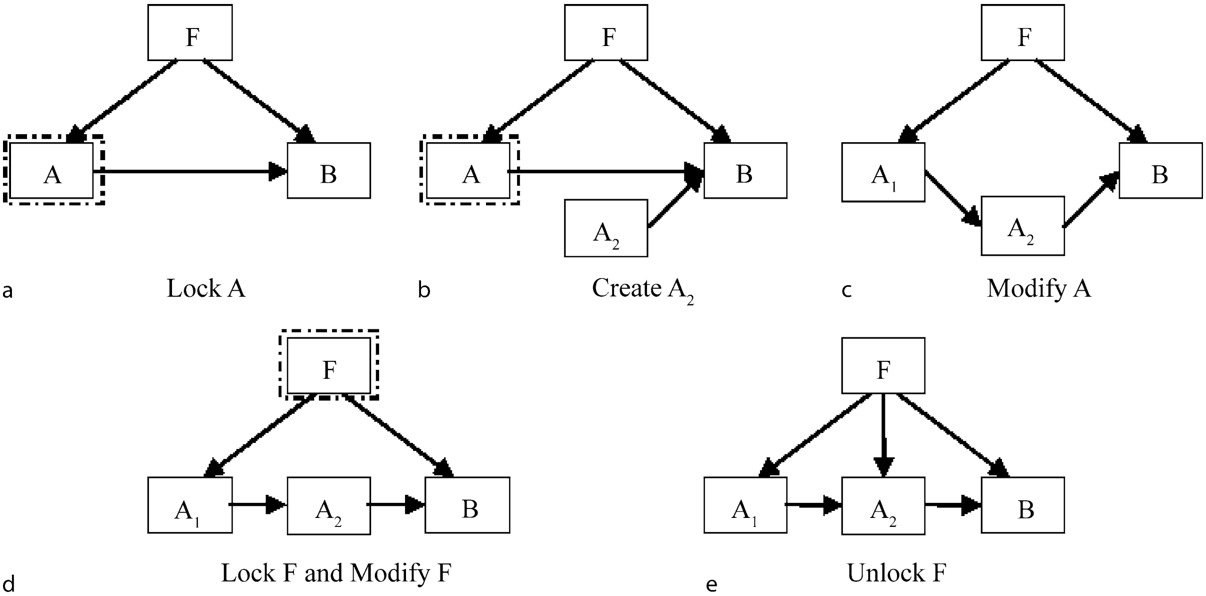
operations (i. e., search, insertion, and deletion) that are committed within each time unit. It is used to measure the efficiency of the concurrency control protocols. The other concern is to prevent phantom access. Phantom access refers to the update operation that occurs before the commitment and in the ranges of a search/deletion operation, while not reflected in the results of that search/deletion operation. The ability to prevent phantom access can be regarded as a certain level of consistency and isolation in ACID rules.

Historical Background

In the last several decades, spatial data access methods have been proposed and developed to manage multidimensional databases as required in GIS, computer-aided design, and scientific modeling and analysis applications. Representative spatial data access methods are R-trees [5], and space-filling curve with B-trees [4]. As shown in Fig. 1a, the R-tree groups spatial objects into Minimum Bounding Rectangles (MBR), and constructs a hierarchical tree structure to organize these MBRs. Differently, Fig. 1b shows the space-filling curve which splits the data space into equal-sized rectangles and uses their particular curve (e. g., Hilbert curve) identifications to index the objects in the cells into one-dimensional access methods, e. g., the B-tree family.

In order to apply the widely studied spatial access methods to real applications, particular concurrency control protocols are required for the multi-user environment. The simultaneous operations on spatial databases need to be treated as exclusive operations without interfering with each other. In other words, the results of any operation have to reflect the current stable snapshot of the spatial database at the commit time.

The concurrency control techniques for spatial databases have to be integrated with spatial access methods to process simultaneous operations. Most of the concurrency control techniques were developed for one-dimensional databases. However, the existing spatial data access methods, such as R-tree family and grid files, are quite different from the one-dimensional data access methods (e. g., overlaps among data objects and among index nodes are allowed). Therefore, the existing concurrency control methods are not suitable for these spatial databases. Furthermore, because the spatial data set usually is not globally ordered, some traditional concurrency control techniques such as link-based protocols are difficult to adapt to spatial databases.



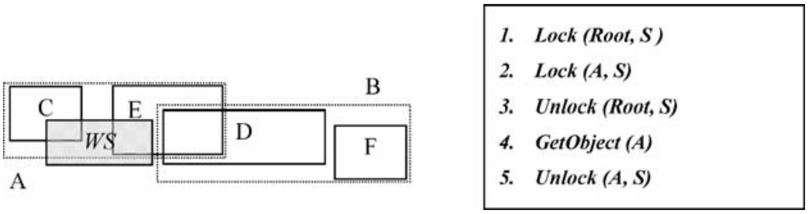
Concurrency Control for Spatial Access Method, Figure 2 Example of Node Split in Link-based Protocol

Spatial Concurrency Control Techniques

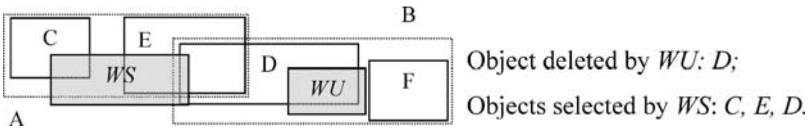
Since the last decade of the 20th century, concurrency control protocols on spatial access methods have been proposed to meet the requirements of multi-user applications. The existing concurrency control protocols mainly focus on the R-tree family, and most of them were developed based on the concurrency protocols on the B-tree family. Based on the locking strategy, these protocols can be classified into two categories, namely, link-based methods and lock-coupling methods.

The link-based methods rely on a pseudo global order of the spatial objects to isolate each concurrent operation. These approaches process update operations by temporally disabling the links to the indexing node being updated so that the corresponding search operations will not retrieve any inconsistent data. For instance, to split node A into A_1 and A_2 in Fig. 2, a lock will be requested to disable the link from A to its right sibling node B (step a) before the actual split is performed. Then, a new node A_2 will be created in step b by using the second half of A, and linked to node B. In step c, A will be modified to be A_1 (by removing the second half), and then unlocked. Node F will be locked before adding a link from F to A_2 in step d. Finally, F will be unlocked in step e, and thus the split is completed. Following this split process, no search operations can access A_2, and no update operations can access $A(A_1)$ before step c. Therefore, the potential confliction caused by concurrent update operations on node A can be prevented. As one example of the link-based approach, R-link tree, a right-link style algorithm [6], has been proposed to protect concurrent operations by assigning logical sequence numbers (LSNs) on the nodes of R-trees. This approach assures each operation has at most one lock at a time. However, when a propagating node splits and the MBR updates, this algorithm uses lock coupling. Also, in this approach, additional storage is required to maintain additional information, e. g., LSNs of associated child nodes. Concurrency on the Generalized Search Tree (CGiST) [7] protects concurrent operations by applying a global sequence number, the Node Sequence Number (NSN). The counter of the NSN is incremented in a node split and a new value is assigned to the original node with the new sibling node receiving the original node's prior NSN and right-link pointer. In order for an insert operation to execute correctly in this algorithm, multiple locks on two or more levels must be held. Partial lock coupling (PLC) [10] has been proposed to apply a link-based technique to reduce query delays due to MBR updates for multi-dimensional index structures. The PLC technique provides high concurrency by using lock coupling only in MBR shrinking operations, which are less frequent than expansion operations.

The lock-coupling-based algorithms [3,8] release the lock on the current node only when the lock on the next node to be visited has been granted while processing search operations. As shown in Fig. 3, using the R-tree in Fig. 1a, suppose objects C, E, D, and F are indexed by an R-tree with two leaf nodes A and B. A search window WS can be processed using the lock-coupling approach. The locking sequence in Fig. 3 can protect this search operation



Concurrency Control for Spatial Access Method, Figure 3 Example of Locking Sequence Using Lock-coupling for WS



Concurrency Control for Spatial Access Method, Figure 4 Example of Phantom Update

from reading the intermediate results of update operations as well as the results of update operations submitted after *WS*. During node splitting and MBR updating, this scheme holds multiple locks on several nodes simultaneously. The dynamic granular locking approach (DGL) has been proposed to provide phantom update protection (discussed later) in R-trees [1] and GiST [2]. The DGL method dynamically partitions the embedded space into lockable granules that can adapt to the distribution of the objects. The lockable granules are defined as the leaf nodes and external granules. External granules are additional structures that partition the non-covered space in each internal node to provide protection. Following the design principles of DGL, each operation requests locks only on sufficient granules to guarantee that any two conflicting operations will request locks on at least one common granule.

Scientific Fundamentals

ACID Rules

Concurrency control for spatial access methods should assure the spatial operations are processed following the ACID rules [9]. These rules are defined as follows.

- Atomicity – Either all or no operations are completed.
- Consistency – All operations must leave the database in a consistent state.
- Isolation – Operations cannot interfere with each other.
- Durability – Successful operations must survive system crashes.

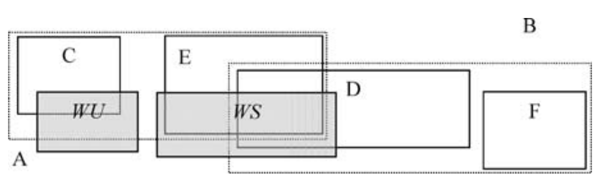
The approaches to guarantee Atomicity and Durability in traditional databases can be applied in spatial databases. Current research on spatial concurrency control approaches mainly focus on the Consistency and Isolation rules. For example, in order to retrieve the valid records, spatial queries should not be allowed to access the intermediate results of location updates. Similarly, the concurrent location updates with common coverage have to be isolated as sequential execution; otherwise, they may not be processed correctly.

Phantom Update Protection

In addition to the ACID rules, phantom update protection is used to measure the effectiveness of a concurrency control. An example of phantom update is illustrated in Fig. 4, where *C*, *E*, *D*, and *F* are objects indexed in an R-tree, and leaf nodes *A*, *B* are their parents, respectively. A deletion with the window *WU* is completed before the commitment of the range query *WS*. The range query returns the set {*C*, *E*, *D*}, even object *D* should have been deleted by *WU*. A solution to prevent phantom update in this example is to lock the area affected by *WU* (which is $D \cup WU$) in order to prevent the execution of *WS*.

Measurement

The efficiency of concurrency control for spatial access methods is measured by the throughput of concurrent spatial operations. The issue to provide high throughput is to reduce the number of unnecessary conflicts among locks. For the example shown in Fig. 5, even if the update operation with window *WU* and the range query with window *WS* intersect with the same leaf node *A*, they will not affect each other's results. Therefore, they should be allowed to access *A* simultaneously. Obviously, the smaller the lockable granules, the more concurrency operations will be allowed. However, this may significantly increase the number of locks in the database, and therefore generate additional overhead on lock maintenance. This is a tradeoff that should be considered when designing concurrency control protocols.



Concurrency Control for Spatial Access Method, Figure 5 Example of Efficient Concurrency Control

Key Applications

Concurrency control for spatial access methods are generally required in commercial multi-dimensional database systems. These systems are designed to provide efficient and reliable data access. Usually, they are required to reliably handle a large amount of simultaneous queries and updates. Therefore, sound concurrency control protocols are required in these systems.

In addition, concurrency control methods are required in many specific spatial applications which have frequent updates or need fresh query results. For instance, a mobile advertise/alarm system needs to periodically broadcast time-sensitive messages to cell phone users within a certain range. Concurrency control methods should be employed to protect the search process from frequent location updates, because the updates are not supposed to reveal their intermediate or expired results to the search process. Another example is a taxi management system that needs to assign a nearest available taxi based on a client's request. Concurrency control methods need to be applied to isolate the taxi location updating and queries so that the query results are consistent to the up-to-date snapshot of the taxi locations.

Future Directions

The study on spatial concurrency control is far behind the research on spatial query processing approaches. There are two interesting and emergent directions in this field. One is to apply concurrency control methods on complex spatial operations; the other is to design concurrency control protocols for moving object applications.

Complex spatial operations, such as spatial join, k-nearest neighbor search, range nearest neighbor search, and reverse nearest neighbor search, require special concern on concurrency control to be applied in multi-user applications. For example, how to protect the changing search range, and how to protect the large overall search range have to be carefully designed. Furthermore, the processing methods of those complex operations may need to be redesigned based on the concurrency control protocol in order to improve the throughput.

Spatial applications with moving objects have attracted significant research efforts. Even though many of these applications assume that the query processing is based on main memory, their frequent data updates require sophisticated concurrency control protocols to assure the correctness of the continuous queries. In this case, concurrency access framework will be required to support the frequent location updates of the moving objects. Frequent update operations usually result in a large number of exclusive locks which may significantly degrade the throughput. Solutions to improve the update speed and reduce the coverage of operations have to be designed to handle this scenario.

Cross References

► Indexing, Hilbert R-tree, Spatial Indexing, Multimedia Indexing

Recommended Reading

1. Chakrabarti, K., Mehrotra, S.: Dynamic Granular Locking Approach to Phantom Protection in R-trees. In: Proceedings of IEEE International Conference on Data Engineering, Orlando, Florida, 23–27 Feb 1998
2. Chakrabarti, K., Mehrotra, S.: Efficient Concurrency Control in Multi-dimensional Access Methods. In: Proceedings of ACM SIGMOD International Conference on Management of Data, Philadelphia, Pennsylvania, 1–3 June 1999
3. Chen, J.K., Huang, Y.F., Chin, Y.H.: A Study of Concurrent Operations on R-Trees. Information Sciences: an International Journal. **98**(1–4), 263–300 (1997)
4. Gaede, V., Gunther, O.: Multidimensional Access Methods. ACM Comput. Surv. **30**(2), 170–231 (1998)
5. Guttman, A.: R-trees: A Dynamic Index Structure for Spatial Searching. In: Proceedings of ACM SIGMOD International Conference on Management of Data, Boston, Massachusetts, 18–21 June 1984
6. Kornacker, M., Banks, D.: High-Concurrency Locking in R-Trees. In: Proceedings of International Conference on Very Large Data Bases, Zurich, Switzerland, 11–15 Sept 1995
7. Kornacker, M., Mohan, C., Hellerstein, J.: Concurrency and Recovery in Generalized Search Trees. In: Proceedings of ACM SIGMOD International Conference on Management of Data, Tucson, Arizona, 13–15 May 1997
8. Ng, V., Kamada, T.: Concurrent Accesses to R-Trees. In: Proceedings of Symposium on Advances in Spatial Databases, Singapore, 23–25 June 1993
9. Ramakrishnan, R., Gehrke, J.: Database Management Systems, 2nd edn. McGraw-Hill, New York, NY, USA (2001)
10. Song, S.I., Kim, Y.H., Yoo, J.S.: An Enhanced Concurrency Control Scheme for Multidimensional Index Structure. IEEE Trans. Knowl. Data. Eng. **16**(1), 97–111 (2004)

Concurrency Control Protocols

► Concurrency Control for Spatial Access
► Concurrency Control for Spatial Access Method

Concurrent Spatial Operations

► Concurrency Control for Spatial Access
► Concurrency Control for Spatial Access Method

Conditional Spatial Regression

► Spatial and Geographically Weighted Regression

Conflation

► Ontology-Based Geospatial Data Integration
► Positional Accuracy Improvement (PAI)

Conflation of Features

SHARAD SETH, ASHOK SAMAL
Department of Computer Science and Engineering,
University of Nebraska-Lincoln, Lincoln, NE, USA

Synonyms

Automated map compilation; Realignment; Rubber-sheeting; Data integration; Entity integration; Vertical conflation; Feature matching

Definition

In GIS, conflation is defined as the process of combining geographic information from overlapping sources so as to retain accurate data, minimize redundancy, and reconcile data conflicts [7]. The need for conflation typically arises in updating legacy data for accuracy or missing features/attributes by reference to newer data sources with overlapping coverage. For example, the street-name and address-range data from the US Census Bureau can be conflated with the spatially accurate USGS digital-line-graph (DLG) to produce a more accurate and useful source than either dataset. Conflating vector GIS data with raster data is also a common problem.

Conflation can take many different forms. Horizontal conflation refers to the matching of features and attributes in adjacent GIS sources for the purpose of eliminating positional and attribute discrepancies in the common area of the two sources. Vertical conflation solves a similar problem for GIS sources with overlapping coverage. As features are the basic entities in a GIS, the special case of feature conflation has received much attention in the published research. The data used for conflation are point, line, and area features and their attributes. Figure 1 illustrates a problem solved by feature conflation. The first two GIS data layers show a digital ortho-photo and a topographic map of the Mall area in Washington D.C. In the third layer on the right, showing an overlay of the two sources, the corresponding features do not exactly line up. With conflation, these discrepancies can be minimized, thus improving the overall accuracy of the data sources.

Historical Background

Until the 1980s, the collection of geographical information in digital form was expensive enough that having multiple sources of data for the same region, as required for conflation, was possible only for large governmental organizations. It is not surprising, therefore, that early use of conflation was initiated by governmental agencies. The applications related to the automation of *map compilation* for transferring positional information from a *base map* to a non-geo-referenced target map. An iterative process, called alignment or rubber-sheeting, was used to bring the coordinates of the two maps into mutual consistency. The latter term alludes to stretching the target map that is printed on a rubber sheet so as to align it with the base map at all points. Although contemplated many years earlier [16], the first semi-automated systems for alignment came into existence only in the mid-1980s. These interactive systems were screen-based and image-driven [8]. The operator was allowed, and even assisted, to select a pair of intersections to be matched. With each additional selected pair, the two maps were brought into closer agreement.

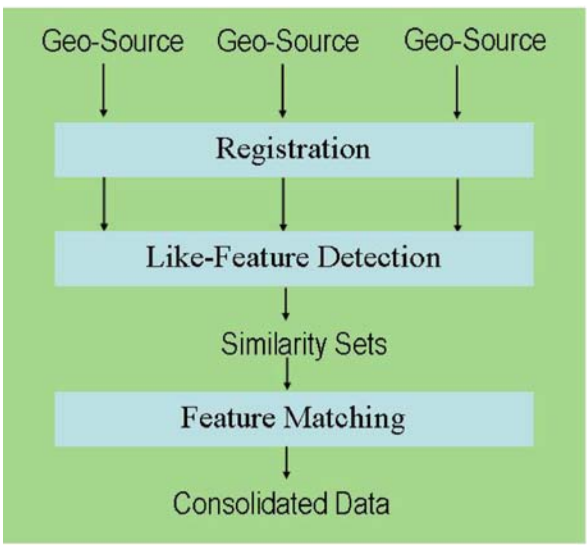
Fully automated systems, later developed in a joint project between the US Geological Society (USGS) and the Bureau of Census, aimed at consolidating the agencies' 5,700 pairs of metropolitan map sheet files [12]. This development was facilitated by parallel advances in computer graphics devices, computational-geometry algorithms, and statistical pattern recognition. Automation of the process required replacing the operator's skills at discerning like features with an analogous feature-matching algorithm on the computer. Alignment may be thought of as a mathematical transformation of one image that preserves topology. A single global transformation may be insufficient to correct errors occurring due to local distortions, thus necessitating local alignments for different regions of the image. Delauney triangulation defined by selected points is preferred for rubber-sheeting because it minimizes the formation of undesirable thin, long triangles.

Early work in feature conflation was based on proximity of features in non-hierarchical ("flattened") data sources. Because GIS data are typically organized into a hierarchy of classes and carry much information that is not position-related, such as, names, scalar quantities, and geometrical shapes, the methods used to discover identical objects can go beyond proximity matches and include rule-based approaches [3], string matching, and shape similarity [13].

Scientific Fundamentals

A prototype of a feature conflation system is shown in Fig. 2. Such a system would typically form the back end of a geographic information and decision-support system used to respond to user queries for matched features. Some

Conflation of Features, Figure 1 Two GIS data layers for Washington DC and their overlay



Conflation of Features, Figure 2 Feature conflation steps

systems may not implement all three steps while others may further refine some of the steps, e. g., like-feature-detection may be split into two steps that either use or ignore the geographical context of the features during comparison. Further details of the basic steps appear in the following sections.

Registration and Rectification

Registration refers to a basic problem in remote sensing and cartography of realigning a recorded digital image with known ground truth or another image. An early survey in geographic data processing [9] formulates the registration problem in remote sensing as follows:

> The scene under observation is considered to be a 2-D intensity distribution $f(x,y)$. The recorded digital, another 2-D distribution $g(u,v)$, is related to the "true" scene $f(x,y)$ through an unknown transformation T:
>
> $$g(u, v) = T(f(x, y)) .$$
>
> Thus, in order to recover the original information from the recorded observations, we must first determine the nature of the transformation T, and then execute the inverse operation T^{-1} on this image.

Often, because only indirect information is available about T, in the form of another image or map of the scene in question, the goal of registration becomes finding a mathematical transformation on one image that would bring it into concurrence with the other image. Geometric distortions in the recorded image, which affect only the position and not the magnitude, can be corrected by a rectification step that only transforms the coordinates.

Like-Feature Detection

The notion of similarity is fundamental to matching features, as it is to many other fields, including, pattern recognition, artificial intelligence, information retrieval, and psychology. While the human view of similarity may be subjective, automation requires objective (quantitative) measures.

Similarity and distance are complementary concepts. It is often intuitively appealing to define a distance function $d(A,B)$ between objects A and B in order to capture their dissimilarity and convert it to a normalized similarity measure by its complement:

$$s(A, B) = 1 - \frac{d(A, B)}{U} , \tag{1}$$

where the normalization factor U may be chosen as the maximum distance between any two objects that can occur in the data set. The normalization makes the value of similarity a real number that lies between zero and one.

Mathematically, any distance function must satisfy the properties if minimality ($d(a,b) \geq d(a,a) \geq 0$), symmetry ($d(a,b) = d(b,a)$), and triangular inequality ($d(a,b) + d(b,c) \geq d(a,c)$). However, in human perception studies, the distance function must be replaced by the "judged distance" for which all of these mathematical axioms have been questioned [14]. Tversky [15] follows a set-theoretic approach in defining similarities between two objects as a function of the attributes that are shared by the two or by one but not the other. His definition is not required to follow any of the metric axioms. It is particularly well suited to fuzzy attributes with discrete overlapping ranges of values.

In GIS, the two objects being compared often have multiple attributes, such as name, location, shape, and area. The name attribute is often treated as a character string for comparison using the well-known Hamming or Levenshtein metric aimed at transcription errors. An alternative measure of string comparison, based on their phonetic representation [6], may be better suited to transcription errors. However, the names are often word phrases that may look very different as character strings, but connote the same object, e. g., "National Gallery of Art" and "National Art Gallery". Table 1 [13] shows the type of string errors that string matching should accommodate:

For locations or points, the Euclidean distance is commonly used for proximity comparison. A generalization to linear features, such as streets or streams, is the Hausdorff distance, which denotes the largest minimum distance between the two linear objects. Goodchild and Hunter [5] describe a less computer-intensive and robust method that relies on comparing two representations with varying accuracy. It estimates the percentage of the total length of the low-accuracy representation that is within a specified distance of the high-accuracy representation.

Conflation of Features, Table 1 Typical string errors and differences that matching should accommodate

Error Type	Examples	
	Sample 1	Sample 2
Word omission	Abraham Lincoln Memorial	Lincoln Memorial
Word substitution	Reagan National Airport	Washington National Airport
Word transposition	National Art Gallery	National Gallery of Art
Word abbreviation	National Archives	Nat'l Archives
Character omission	Washingtn Monument	Washington Monument
Character substitution	Frear Gallery	Freer Gallery

The shape is an important attribute of polygonal features in GIS, such as building outlines and region boundaries. As polygons can be regarded as linear features, the Goodchild and Hunter approach may be adapted to define shape comparison. A two-step process is described for this purpose by Samal et al. [13]. First, a veto is imposed if the aspect ratios are significantly different. Otherwise, the shapes are scaled to match the lengths of their major axes and overlaid by aligning their center points. The similarity of a less accurate shape A to a more accurate shape B is the percentage A within the buffer zone of B (see Fig. 3). When the accuracy of the two sources is comparable, the measure could be taken as the average value of the measure computed both ways.

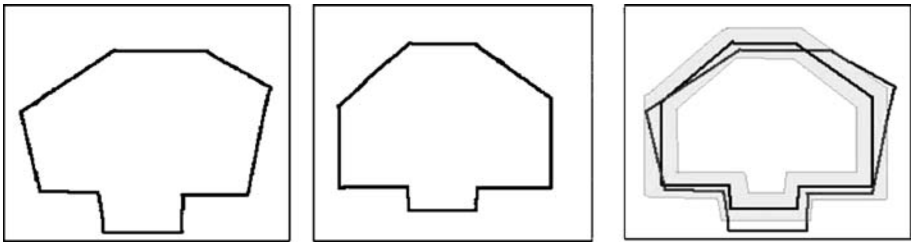
Comparing scalars (reals or integers) seems to be straightforward: take the difference as their distance and convert to similarity by using Eq. (1). The normalization factor U, however, must be chosen carefully to match intuition. For example, one would say that the pair of numbers 10 and 20 is less similar to the pair 123,010 and 123020, even though the difference is the same in both cases. Hence, the normalization factor should be equated with the magnitude of the range of values defined for scalars a and b.

The hierarchical nature of GIS data makes it possible to also assess the similarities of two objects along their categorical structure. A knowledge base of spatial categories can be built using Wordnet [4] and Spatial Data Transfer Standard (SDTS) [11].

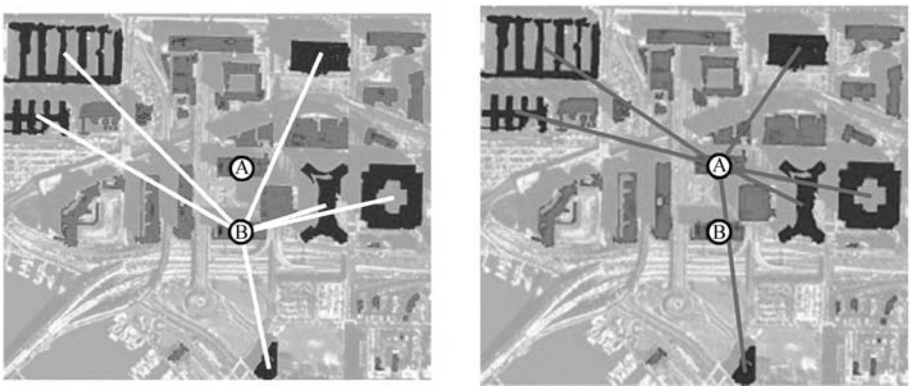
Context

Clearly, context plays an important role in the human perception of similarity. The similarity measures described above, however, are *context-independent*: two features are compared in isolation without reference to other features on their respective sources. Context-independent similarity measures alone are not always sufficient enough to determine feature matches unambiguously, necessitating the use of some form of context to resolve such cases.

The *geographic context* is defined as the spatial relationships between objects in an area [13]. Examples of such relationships include topologies, distances, and directions. Topological relationships, such as disjoint, meet, overlap, and covering are used by the researchers at the University of Maine to model the context of areal features in applications involving query by sketch and similarity of spatial scenes [2]. Distances and angles of a feature to other features have also been used to represent the geographic con-



Conflation of Features, Figure 3 Two polygons and their buffered intersection



Conflation of Features, Figure 4 Two similar features and their geographic contexts

text [13]. Figure 4 shows the geographic contexts of two nearby features with similar shapes. The contexts can be seen to be different enough to disambiguate these two features when they are compared with a candidate feature in another source. Further, to keep the cost of context-dependent matching under control, it may be enough to define the geographic context with respect to only a small number of well chosen landmark features.

Feature Matching

The similarity measures discussed above for individual attributes of a feature must be combined in some fashion to provide overall criteria for feature matching. According to Cobb et al. [3], "The assessment of feature match criteria is a process in which evidence must be evaluated and weighed and a conclusion drawn – not one in which equivalence can be unambiguously determined … after all, if all feature pairs matched exactly, or deviated uniformly according to precise processes, there would be no need to conflate the maps!"

The problem can be approached as a restricted form of the classification problem in pattern recognition: Given the evidence provided by the similarity scores of different attributes of two features, determine the likelihood of one feature belonging to the same class as the other feature. Because of this connection, it is not surprising that researchers have used well-known techniques from pattern recognition to solve the feature matching problem. These techniques include clustering [1] and fuzzy logic [17].

For example, similarity of two buildings appearing in different GIS data layers (as in Fig. 1a and b) could be established by comparing their individual attributes, such as shape and coordinates. These context-independent measures, however, may not be sufficient and it may become necessary to use the geographical context to resolve ambiguities or correct errors.

Key Applications

Coverage Consolidation: Data gathering is the most expensive part of building a geographical information system (GIS). In traditional data gathering, this expense is directly related to the standards of rigor used in data collection and data entry. Feature conflation can reduce the cost of GIS data acquisition by combining inexpensive sources into a superior source. With the widespread use of the Web and GPS, the challenge in consolidation is shifting from improving accuracy to integrating an abundance of widely distributed sources by automated means.

Spatial Data Update: By identifying common and missing features between two sources through feature conflation, new features can be added to an old source or their attributed updated from a newer map.

Coverage Registration: Non-georeferenced spatial data must be registered before it can be stored in a GIS. Good registration requires choosing a number of features for which accurate geo-positional information is available and which are also spatially accurate on the source. Spatial data update can help in identifying good candidate features for registration.

Error Detection: Feature conflation can not only tell which features in two sources are alike, but also provide a degree of confidence for these assertions. The pairs with low confidence can be checked manually for possible errors.

Future Directions

Conflation in GIS can be thought of as part of the broader problems in the information age of searching, updating, and integration of data. Because the sense of place plays such an important role in our lives, all kinds of nongeographical data related to history and culture can be tied to a place and thus become a candidate for conflation. In this view, geographical reference becomes a primary key used by search engines and database applications to consolidate, filter, and access the vast amount of relevant data distributed among many data sources. The beginnings of this development can already be seen in the many applications already in place or envisaged for Google Earth and other similar resources. If the consolidated data remains relevant over a period of time and finds widespread use, it might be stored and used as a new data source, much in the same fashion as the results of conflation are used today.

The traditional concern in conflation for positional accuracy will diminish in time with the increasing penetration of GPS in consumer devices and the ready availability of the accurate position of all points on the earth. The need for updating old data sources and integrating them with new information, however, will remain an invariant.

Cross References

► Conflation of Geospatial Data
► Geospatial Semantic Integration
► Ontology-Based Geospatial Data Integration

Recommended Reading

1. Baraldi, A., Blonda, P.: A survey of fuzzy clustering algorithms for pattern recognition. IEEE Transactions on Systems, Man and Cybernetics Parts I and II **29**(6), 778–785 (1999)
2. Bruns, H., Eggenhofer, M.: Similarity of Spatial Scenes. In: Molenaar, M., Kraak, M.J. (eds.) Proc. 7th Int. Symposium on Spatial Data Handling, pp. 31–42. Taylor and Francis, London (1996)
3. Cobb, M., Chung, M.J., Foley III, H., Petry, F.E., Shaw, K.B.: A Rule-based Approach for the Conflation of Attributed Vector Data. Geoinformatica **2**(1), 7–35 (1998)
4. Fellbaum, C. (ed.): Wordnet: An Electronic Lexical Database. MIT Press, Cambridge, MA (1998)
5. Goodchild, M.F., Hunter, G.J.: A Simple Positional Accuracy Measure for Linear Features. Int. J. Geogr. Inf. Sci. **11**(3), 299–306 (1997)
6. Hall, P., Dowling, G.:Approximate String Matching. ACM Computing Surveys **12**(4), 381–402 (1980)
7. Longley, P.A., Goodchild, M.F., Maguire, D.J., Rhind, D.W.: Geographic Information Systems and Science. John Wiley & Sons, Chichester, West Sussex, England (2001)
8. Lynch, M.P., Saalfield, A.: Conflation: Automated Map Compilation – A Video Game Approach. In: Proc. Auto-Carto 7 ACSM/ASP, Falls Church, VA, 11 Mar 1985
9. Nagy, G., Wagle, S.: Geographic Data Processing. ACM Computing Surveys **11**(2), 139–181 (1979)
10. Rodriguez, A., Egenhofer, M.: Assessing Similarity Among Geospatial Feature Class Definitions. In: Vckorski, A., Brasel, K.E., Schek, H.J. (eds.) Lecture Notes in Computer Science 1580, pp. 189–202. Springer, Berlin (1999)
11. Rodriguez, A., Egenhofer, M.: Determining Semantic Similarity Among Entity Classes from Different Ontologies. IEEE Transactions on Knowledge and Data Engineering **15**, 442–456 (2003)
12. Saalfield, A.: Conflation: Automated Map Compilation. Int. J. GIS, **2**(3), 217–228 (1988)
13. Samal, A., Seth, S., Cueto, K.: A feature-based approach to conflation of geospatial sources. Int. J. GIS **18**(5), 459–589 (2004)
14. Santini, S., Jain, R.: Similarity Measures. IEEE Trans. Pattern Analysis and Machine Intelligence **21**(9), 87–883 (1999)
15. Tversky, A.: Features of similarity. Psychological Review **84**, 327–352 (1977)
16. White, M.: The Theory of Geographical Data Conflation. Internal Census Bureau draft document, (1981)
17. Zadeh, L.A.: Fuzzy Sets. Information and Control **8**, 338–353 (1965)

C

Conflation of Geospatial Data

CHING-CHIEN CHEN[1], CRAIG A. KNOBLOCK[2]
[1] Geosemble Technologies, El Segundo, CA, USA
[2] Department of Computer Science, University of Southern California, Marina del Rey, CA, USA

Synonyms

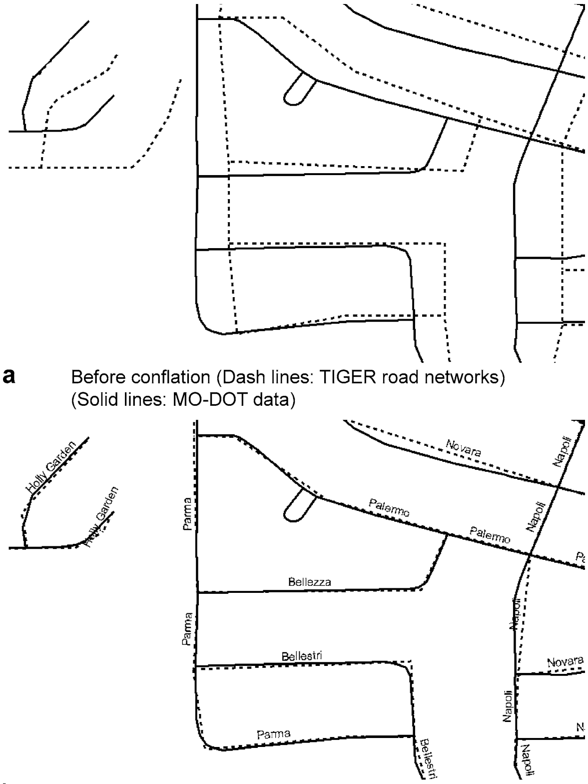
Geospatial data alignment; Geospatial data reconciliation; Computer cartography; Imagery conflation

Definition

Geospatial data conflation is the compilation or reconciliation of two different geospatial datasets covering overlapping regions [1]. In general, the goal of conflation is to combine the best quality elements of both datasets to create a composite dataset that is better than either of them. The consolidated dataset can then provide additional information that cannot be gathered from any single dataset.

Based on the types of geospatial datasets dealt with, the conflation technologies can be categorized into the following three groups:

- Vector to vector data conflation: A typical example is the conflation of two road networks of different accu-



a Before conflation (Dash lines: TIGER road networks) (Solid lines: MO-DOT data)

b After conflation (Road names are from the TIGER road network)

Conflation of Geospatial Data, Figure 1 An example of vector to vector conflation

a MO-DOT data with USGS imagery before conflation

b After conflation

Conflation of Geospatial Data, Figure 2 An example of vector to raster data conflation (modified figure from [8])

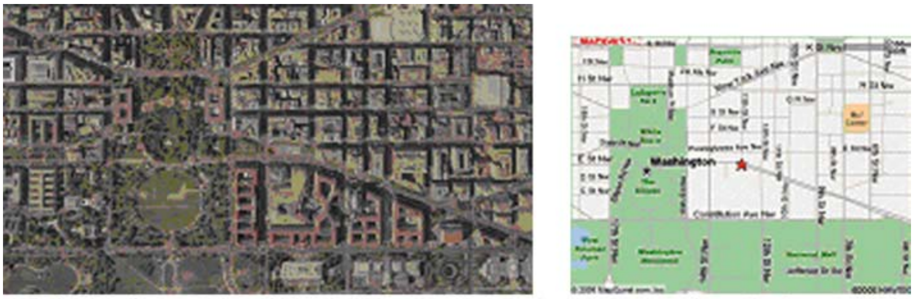
racy levels. Figure 1 shows a concrete example to produce a superior dataset by integrating two road vector datasets: road network from US Census TIGER/Line files, and road network from the department of transportation, St. Louis, MO (MO-DOT data).

- Vector to raster data conflation: Fig. 2 is an example of conflating a road vector dataset with a USGS 0.3 m per pixel color image. Using the imagery as the base dataset for position, the conflation technique can correct the vector locations and also annotate the image with appropriate vector attributes (as Fig. 2b).
- Raster to raster data conflation: Fig. 3 is an example of conflating a raster street map (from MapQuest) with a USGS image. Using the imagery as the base dataset for position, the conflation technique can create intelligent images that combine the visual appeal and accuracy of imagery with the detailed attributes often contained in maps (as Fig. 3b).

Also note that although the examples shown in Fig. 1 to 3 are the conflation of datasets covering the same region (called vertical conflation), the conflation technologies can also be applied to merge adjacent datasets (called horizontal conflation).

Historical Background

For a number of years, significant manual effort has been required to conflate two geospatial datasets by identifying features in two datasets that represent the same real-world features, then aligning spatial attributes and non-spatial attributes of both datasets. Automated vector and vector conflation was first proposed by Saalfeld [1], and the initial focus of conflation was using geometrical similarities between spatial attributes (e. g., location, shape, etc.) to eliminate the spatial inconsistency between two overlapping vector maps. In particular, in [1], Saalfeld discussed mathematical theories to support the automatic process. From then, various vector to vector conflation techniques have been proposed [2,3] and many GIS systems



a Before conflation (left: a USGS 0.3m/p color image;right: a street map from MapQuest)

b After conflation

Conflation of Geospatial Data, Figure 3 An example of raster map to imagery conflation (modified figure from [10])

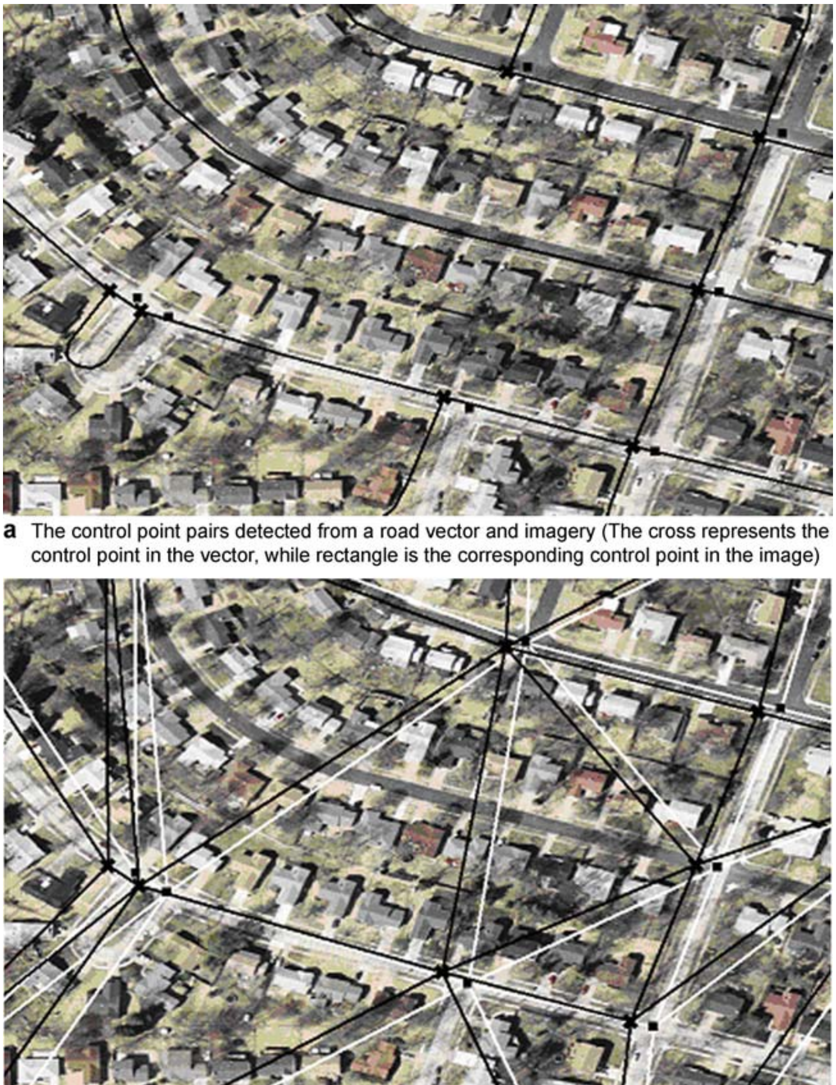
(such as Conflex[1]) have been implemented to achieve the alignments of geospatial datasets. More recently, with the proliferation of attributed vector data, attribute information (i. e., non-spatial information) has become another prominent feature used in the conflation systems, such as ESEA MapMerger[2] and the system developed by Cobb et al. [4]. Most of the approaches mentioned above focus on vector to vector conflation by adapting different techniques to perform the matching. However, due to the rapid advances in remote sensing technology from the 90s to capture high resolution imagery and the ready accessibility of imagery over the Internet, such as Google Maps[3] and Microsoft TerraService[4], the conflation with imagery (such as *vector to imagery* conflation, *imagery to imagery* conflation and *raster map to imagery* conflation) has become one of the central issues in GIS. The objectives of these imagery-related conflation are, of course, to take full advantages of updated high resolution imagery to improve out-of-date GIS data and to display the ground truth in depth with attributes inferred from other data sources (as the examples shown in Fig. 2b and Fig. 3b). Due to the natural characteristics of imagery (or, more generally, geospatial raster data), the matching strategies used in conflation involve more image-processing or pattern recognition technologies. Some proposed approaches [5,6] rely on edge detections or interest-point detections to extract and convert features from imagery to vector formats, and then apply *vector to vector* conflation to align them. Other approaches [7,8,9], however; utilize the existing vector data as prior knowledge to perform a vector-guided image processing. Conceptually, the spatial information on the vector data represents the existing knowledge about the approximate location and shape of the counterpart elements in the image, thus improving the accuracy and running time to detect matched features from the image. Meanwhile, there are also numerous research activities [10,11,12] focusing on conflating different geospatial raster datasets. Again, these approaches perform diverse image-processing techniques to detect and match counterpart elements, and then geometrically align these raster datasets so that the respective pixels or their derivatives (edges, corner point, etc.) representing the same underlying spatial structure are fused.

Today, with the popularity of various geospatial data, automatic geospatial data conflation is rather an area of active research. Consequently, there are various commercial products, such as MapMerger and Conflex, supporting automatic *vector to vector* data conflation with limited human intervention. However, there are no commercial products to provide automatic *vector to raster* or *raster to raster* conflation.

Scientific Fundamentals

A geospatial data conflation system requires efficient and robust geometric and statistical algorithms, and image pro-

[1] http://www.digitalcorp.com/conflex.htm
[2] http://www.esea.com/products/
[3] http://maps.google.com/
[4] http://terraservice.net/



a The control point pairs detected from a road vector and imagery (The cross represents the control point in the vector, while rectangle is the corresponding control point in the image)

b Delauny triangulation (Black triangle: Delaunay triangulation based on detected control points on road vector. White triangle: Corresponding Delaunay triangulation based on detected control points on image)

Conflation of Geospatial Data, Figure 4
An example of Delaunay triangulation based on control points (modified figure from [8])

cessing and pattern recognition techniques to implement a rather broad spectrum of mathematical theories. The framework of conflation process can be generalized into the following steps: 1) Feature matching: Find a set of conjugate point pairs, termed control point pairs, in two datasets, 2) Match checking: Filter inaccurate control point pairs from the set of control point pairs for quality control, and 3) Spatial attribute alignment: Use the accurate control points to align the rest of the geospatial objects (e. g., points or lines) in both datasets by using space partitioning techniques (e. g., triangulation) and geometric interpolation techniques.

During the late 1980s, Saalfeld [1] initialized the study to automate the conflation process. He provided a broad mathematical context for conflation theory. In addition, he proposed an iterative conflation paradigm based on the above-mentioned conflation framework by repeating the matching and alignment, until no further new matches are identified. In particular, he investigated the techniques to automatically construct the influence regions around the control points to reposition other features into alignment by appropriate local interpolation (i. e., to automate the third step in the above-mentioned conflation framework). The conclusion of Saalfeld's work is that Delaunay triangulation is an effective strategy to partition the domain space into triangles (influence regions) to define local adjustments (see the example in Fig. 4). A Delaunay triangulation is a triangulation of the point set with the property that no point falls in the interior of the circumcircle of any triangle (the circle passing through the three triangle vertices). The Delaunay triangulation maximizes the minimum angle of all the angles in the triangulation, thus avoiding elongated, acute-angled triangles. The triangle vertices (i. e., control points) of each triangle define the local transformation within each triangle to reposition other features. The local transformation used for positional interpolation is often the affine transformation, which consists of a linear transformation (e. g., rotation and scaling) followed by a translation. An affine transformation can preserve collinearity and topology. The well-known technique, rubber-sheeting (imagine stretching a dataset as if it were made of rubber), typically refers to the process comprising triangle-based space partition and the transformation of features within each triangle.

What Saalfeld discovered had a profound impact upon conflation techniques. From then on, the rubber-sheeting technique (with some variants) is widely used in conflation algorithms, because of the sound mathematical theories and because of its success in many practical examples. In fact, these days, most of commercial conflation products support the piecewise rubber-sheeting. Due to the fact that rubber-sheeting has become commonly known strategy to geometrically merge datasets based on the control points, many algorithms have been invented around this conflation paradigm with a major focus on solving the matching (correspondence) problem to find accurate control point pairs (i. e., to automate the first two steps in the above-mentioned conflation framework). However, feature matching algorithms differ with the types of datasets undergoing the match operation. In the following, we discuss existing conflation (matching) technologies based on the types of geospatial datasets dealt with.

- ***Vector to vector* conflation:** There have been a number of efforts to automatically or semi-automatically accomplish vector to vector conflation. Most of the existing vector-vector conflation algorithms are with a focus on road vector data. These approaches are different, because of the different methods utilized for locating the counterpart elements from both vector datasets. The major approaches include:
  - Matching vector data based on the similarities of geometric information (such as nodes and lines) [1,2,3].
  - Matching attribute-annotated vector data based on the similarities of vector shapes as well as the semantic similarities of vector attributes [4].
  - Matching vector data with unknown coordinates based on the feature point (e. g., the road intersection) distributions [13].
- ***Vector to imagery* conflation:** *Vector to imagery* (and *Vector to raster*) conflation, on the other hand, mainly focus on developing effective and efficient image processing techniques to resolve the correspondence problem. The major approaches include:
  - Detecting all salient edges from imagery and then comparing with vector data [5].
  - Utilizing vector data to identify corresponding image edges based on (modified) Snakes algorithm [7,14].
  - Utilizing stereo images, elevation data and knowledge about the roads (e. g., parallel-lines and road marks) to compare vector and imagery [9].
  - Exploiting auxiliary spatial information (e. g., the coordinates of imagery and vector, the shape of roads around intersections, etc.) and non-spatial information (e. g., the image color/resolution and road widths) to perform a localized image processing to compute the correspondence [8,15]. Figure 2b is the example result based on this technology.
- ***Raster to raster* conflation:** In general, *raster to raster* conflation (e. g., *imagery to imagery* conflation *and map to imagery* conflation) requires more data-specific image processing techniques to identify the corresponding features from raster data. Some exiting approaches, for example, include:
  - Conflating two images by extracting and matching various features (e. g., edges and feature points) across images [11,12].
  - Conflating a raster map and imagery by computing the relationship between two feature point sets detected from the datasets [10]. In this approach, especially, these feature points are generated by exploiting auxiliary spatial information (e. g., the coordinates of imagery, the orientations of road segments around intersections from the raster map, etc.) and non-spatial information (e. g., the image resolution and the scale of raster maps). Figure 3b is the example result based on this technology.

Key Applications

Conflation technologies are used in many application domains, most notably the sciences and domains using high quality spatial data such as GIS.

Cartography

It is well known that computers and mathematical methods have had a profound impact upon cartography. There has been a massive proliferation of geospatial data, and no longer is the traditional paper map the final product. In fact, the focus of cartography has shifted from map production to the presentation, management and combination of geospatial data. Maps can be produced on demand for specialized purposes. Unfortunately, the data used to produce maps may not always be consistent. Geospatial data conflation can be used to address this issue. For example, we can conflate to out-of-date maps with up-to-date imagery to identify inconsistencies.

GIS

Geographic information provides the basis for many types of decisions ranging from economic and community planning, land and natural resource management, health, safety and military services. Improved geographic data should lead to better conclusions and better decisions. In general, superior data would include greater positional accuracy, topological consistency and abundant attribution information. Conflation technology, of course, plays a major role in producing high quality data for various GIS applications requiring high-quality spatial data.

Computational Geometry

Although originally the conflation technology is intended for consolidating geospatial datasets that are known to contain the same features, the methods employed in conflation can be adapted for other applications. For example, the variants of rubber-sheeting techniques are widely used to support general spatial interpolation. The point or line matching algorithms, in turn, can be used in a broad spectrum of geometric object comparisons.

Aerial Photogrammetry

With the wide availability of high resolution aerial photos, there is a pressing need to analyze aerial photos to detect changes or extract up-to-date features. In general, the problem of extracting features from imagery has been an area of active research for the last 25 years and given the current state-of-the-art will be unlikely to provide near-term fully-automated solutions to the feature extraction problem. For many regions, there are detailed feature datasets that have already been constructed, but these may need to be conflated with the current imagery. Conflating and correlating vector data with aerial photos is more likely to succeed over a pure feature extraction approach since we are able to exploit significant prior knowledge about the properties of the features to be extracted from the photos. Furthermore, after conflation, the attribution information contained in vector dataset can be used to annotate spatial objects to better understand the context of the photos.

Homeland Security

The conflation of geospatial data can provide insights and capabilities not possible with individual data. It is important to the national interest that this automatic conflation problem be addressed since significant statements concerning the natural resources, environment, urban settlements, and particularly internal or Homeland Security, are dependent on the results of accurate conflation of geospatial datasets such as satellite images and geospatial vector data including transportation, hydrographic and cadastral data.

Military Training and Intelligence

Many military training and preparation systems require high quality geospatial data for correctly building realistic training environments across diverse systems/applications. An integrated view of geographic datasets (especially satellite imagery and maps) can also help military intelligence analysts to more fully exploit the information contained in maps (e. g., road/railroad networks and textual information from the map, such as road names and gazetteer data) for analyzing imagery (i. e., identify particular targets, features, and other important geographic characteristics) and use the information in imagery to confirm information in maps. The geospatial data conflation technique is the key technology to accomplish this.

Crisis Management

In a crisis, such as a large fire, a category 5 hurricane, a dirty bomb explosion, emergency personnel must have access to relevant geographic information quickly. Typically, geographic data, such as maps and imagery are important data sources for personnel who are not already familiar with a local area. The conflation technology enables emergency personnel to rapidly integrate the maps, vector data, and imagery for a local area to provide an integrated geographic view of an area of interest.

Transportation Data Update

Many GIS applications require the road vector data for navigation systems. These days, up-to-date high resolution imagery is often utilized to verify and update road vector data. The ability to automatically conflate the original road vector data with images supports more efficient and accurate updates of road vector.

Real Estate

With the growth of the real estate market, there are many online services providing real estate records by superim-

a Before conflation b After conflation

Conflation of Geospatial Data, Figure 5 An example of parcel vector data to imagery conflation

posing the parcel boundaries on top of high-resolution imagery to show the location of parcels on imagery. However, as is typically the case in integrating different geospatial datasets, a general problem in combining parcel vector data with imagery from different sources is that they rarely align (as shown in Fig. 5a). These displacements can mislead the interpretation of parcel and land use data. As the example shown in Fig. 5, parcel data are often represented as polygons and include various attributes such as ownership information, mailing address, acreage, market value and tax information. The cities and counties use this information for watershed and flood plain modelling, neighborhood and transportation planning. Furthermore, various GIS applications rely on parcel data for more accurate geocoding. By conflating parcel vector data and imagery, the detailed attribution information provided by the parcel data (as an example shown Fig. 5b) can be combined with the visible information provided by the imagery. Therefore, the conflation of these datasets can provide cost savings for many applications, such as county, city, and state planning, or integration of diverse datasets for more accurate address geocoding or emergency response.

Future Directions

With the rapid improvement of geospatial data collection techniques, the growth of Internet and the implementation of Open GIS standards, a large amount of geospatial data is now readily available. There is a pressing need to combine these datasets together using conflation technology. Although there has been significant progress on automatic conflation technology in the last few years, there is still much work to be done. Important research problems include, but are not limited to the following: 1) resolving discrepancies between datasets with very different levels of resolution and thematic focus, 2) extending existing technologies to handle a broad range of datasets (in addition to road networks), such as elevation data and hydrographic data, 3) allowing for uncertainty in the feature matching stage, and 4) improving the processing time (especially for raster data) to achieve conflation on the fly.

Cross References

► Change Detection
► Intergraph: Real Time Operational Geospatial Applications
► Photogrammetric Applications
► Uncertain Environmental Variables in GIS
► Voronoi Diagram (Voronoi diagram is the dual of Delaunay Triangulation)

Recommended Reading

1. Saalfeld, A.: Conflation: Automated Map Compilation. Int. J. Geogr. Inf. Sci. **2**(3), 217–228 (1988)
2. Walter, V., Fritsch, D.: Matching Spatial Data Sets: a Statistical Approach. Int. J. Geogr. Inf. Sci. **5**(1), 445–473 (1999)
3. Ware, J.M., Jones, C.B.: Matching and Aligning Features in Overlayed Coverages. In: Proceedings of the 6th ACM Symposium on Geographic Information Systems. Washingnton, D.C. (1998)

4. Cobb, M., Chung, M.J., Miller, V., Foley, H. III, Petry, F.E., Shaw, K.B.: A Rule-Based Approach for the Conflation of Attributed Vector Data. GeoInformatica **2**(1), 7–35 (1998)
5. Filin, S., Doytsher, Y.: A Linear Conflation Approach for the Integration of Photogrammetric Information and GIS Data. Int. Arch. Photogramm. Remote Sens. **33**, 282–288 (2000)
6. Flavie, M., Fortier, A., Ziou, D., Armenakis, C., Wang, S.: Automated Updating of Road Information from Aerial Images. In: Proceedings of American Society Photogrammetry and Remote Sensing Conference. Amsterdam, Holland (2000)
7. Agouris, P., Stefanidis, A., Gyftakis, S.: Differential Snakes for Change Detection in Road Segments. Photogramm. Eng. Remote Sens. **67**(12), 1391–1399 (2001)
8. Chen, C.-C., Knoblock, C.A., Shahabi, C.: Automatically conflating road vector data with orthoimagery. Geoinformatica **10**(4), 495–530 (2006)
9. Eidenbenz, C., Kaser, C., Baltsavias, E.: ATOMI – Automated Reconstruction of Topographic Objects from Aerial Images using Vectorized Map Information. Int. Arch. Photogramm. Remote Sens. **33**(Part 3/1), 462–471 (2000)
10. Chen, C.-C., Knoblock, C.A., Shahabi, C., Chiang, Y.-Y., Thakkar, S.: Automatically and Accurately Conflating Orthoimagery and Street Maps. In: Proceedings of the 12th ACM International Symposium on Advances in Geographic Information Systems. Washington, D.C., USA (2004)
11. Seedahmed, G., Martucci, L.: Automated Image Registration Using Geometrical Invariant Parameter Space Clustering (GIPSC). In: Proceedings of the Photogrammetric Computer Vision. Graz, Austria (2002)
12. Dare, P., Dowman, I.: A new approach to automatic feature based registration of SAR and SPOT images. Int. Arch. Photogramm. Remote Sens. **33**(B2), 125–130 (2000)
13. Chen, C.-C., Shahabi, C., Knoblock, C.A., Kolahdouzan, M.: Automatically and Efficiently Matching Road Networks with Spatial Attributes in Unknown Geometry Systems. In: Proceedings of the Third Workshop on Spatio-temporal Database Management (co-located with VLDB2006). Seoul, Korea (2006)
14. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Int. J. Comput. Vision. **1**(4), 321–331 (1987)
15. Chen, C.-C., Shahabi, C., Knoblock, C.A.: Utilizing Road Network Data for Automatic Identification of Road Intersections from High Resolution Color Orthoimagery. In: Proceedings of the Second Workshop on Spatio-temporal Database Management (co-located with VLDB2004). Toronto, Canada (2004)

Conflict Resolution

► Computing Fitness of Use of Geospatial Datasets
► Smallworld Software Suite

Consequence Management

► Emergency Evacuations, Transportation Networks

Conservation Medicine

► Exploratory Spatial Analysis in Disease Ecology

Constraint Database Queries

LIXIN LI
Department of Computer Sciences,
Georgia Southern University, Statesboro, GA, USA

Synonyms

Constraint query languages; Datalog, SQL; Logic programming language

Definition

A database query language is a special-purpose programming language designed for retrieving information stored in a database. SQL (Structured Query Language) is a very widely used commercially marketed query language for relational databases. Different from conventional programming languages such as C, C++ or Java, a SQL programmer only need to specify the properties of the information to be retrieved, but not the detailed algorithm required for retrieval. Because of this property, SQL is said to be *declarative*. In contrast, conventional programming languages are said to be *procedural*.

To query spatial constraint databases, any query language can be used, including SQL. However, Datalog is probably the most popularly used rule based query language for spatial constraint databases because of its power of recursion. Datalog is also declarative.

Historical Background

The Datalog query language is based on logic programming language Prolog. The history of Datalog queries and logic programming is discussed in several textbooks such as [9,11,12]. Early work on constraint logic programming has been done by Jaffar and Lassez [2]. The concepts of constraint data model and query language have been explored by Kanellakis, Kuper and Revesz [3,4]. Recent books on constraint databases are [5] and [10].

Scientific Fundamentals

A Datalog query program consists of a set of rules of the following form [10]:

$$R_0(x_1, \ldots, x_k) : -R_1(x_{1,1}, \ldots, x_{1,k}), \ldots \quad R_n(x_{n,1}, \ldots, x_{n,k}).$$

where each R_i is either an input relation name or a defined relation name, and the xs are either variables or constants.

Query 1 *For the ultraviolet radiation example in Scientific Fundamentals in Entry* ► Constraint Databases and

Constraint Database Queries, Table 1 *GROUND (x, y, t, i)* using IDW

X	*Y*	*T*	*I*	
x	y	t	i	$2x - y - 20 < q0, 12x + 7y - 216 < q0, 13y + 7t - 286 < q0, 2y - 3t - 12 < q0, y < q15,$
				$((x-2)^2 + (y-14)^2)0.9 + ((x-2)^2 + (y-1)^2)0.5 = (2(x-2)^2 + (y-14)^2 + (y-1)^2)r,$
				$((y-13)^2 + (t-22)^2)60 + (y^2 + (t-1)^2)20 = ((y-13)^2 + (t-22)^2 + y^2 + (t-1)^2)u,$
				$i = u(1-r)$
x	y	t	i	$2x - y - 20 \geq 0, 12x + 7y - 216 < q0, 13y + 7t - 286 < q0, 2y - 3t - 12 < q0, y < q15,$
				$((x-25)^2 + (y-1)^2)0.9 + ((x-2)^2 + (y-1)^2)0.8 = (2(y-1)^2 + (x-25)^2 + (x-2)^2)r,$
				$((y-13)^2 + (t-22)^2)60 + (y^2 + (t-1)^2)20 = ((y-13)^2 + (t-22)^2 + y^2 + (t-1)^2)u,$
				$i = u(1-r)$
x	y	t	y	$\dots$
				$\dots$
x	y	t	i	$2x - y - 20 < q0, 12x + 7y - 216 \geq 0, y \geq 15, y + 3t - 54 < q0, 7y - t - 136 < q0, 2y + 5t - 60 \geq 0,$
				$((x-25)^2 + (y-14)^2)0.5 + ((x-2)^2 + (y-14)^2)0.3 = (2(y-14)^2 + (x-25)^2 + (x-2)^2)r$
				$((y-29)^2 + t^2)20 + ((y-13)^2 + (t-22)^2)40 = ((y-29)^2 + t^2 + (y-13)^2 + (t-22)^2)u,$
				$i = u(1-r)$

Data Interpolation, *find the amount of ultraviolet radiation for each ground location (x,y) at time t.*

Since the input relations in Tables 1 and 2 in Entry "Constraint Databases and Data Interpolation" only record the incoming ultraviolet radiation u and filter ratio r on a few sample points, these cannot be used directly to answer the query. Therefore, to answer this query, the interpolation results of *INCOMING(y,t,u)* and *FILTER(x,y,r)* are needed. To write queries, it is not necessary to know precisely what kind of interpolation method is used and what are the constraints used in the representation interpolation. The above query can be expressed in Datalog as follows [6]:

$$
\begin{aligned}
GROUND(x, y, t, i) :& -INCOMING(y, t, u)\,, \\
& FILTER(x, y, r)\,, \\
& i = u(1 - r)\,.
\end{aligned}
$$

The above query could be also expressed in SQL. Whatever language is used, it is clear that the evaluation of the above query requires a join of the *INCOMING* and *FILTER* relations. Unfortunately, join operations are difficult to express in simple GIS systems, including the ArcGIS system. However, join processing is very natural in constraint database systems.

If the IDW interpolation is used (see Sect. 3 "Key Applications", Entry "Constraint Databases and Data Interpolation"), the final result of the Datalog query, *GROUND(x, y, t, i)*, can be represent by Table 1. Since there are five second-order Voronoi regions for *Incoming* and four regions for *Filter*, as shown in Figs. 8 and 9 in Entry "Constraint Databases and Data Interpolation", there should be twenty tuples in *GROUND(x, y, t, i)* in Table 1. Note that the constraint relations can be easily joined by taking the conjunction of the constraints from each pair of tuples of the two input relations. Finally, in a constraint database system the constraint in each tuple are automatically simplified by eliminating the unnecessary variables u and r.

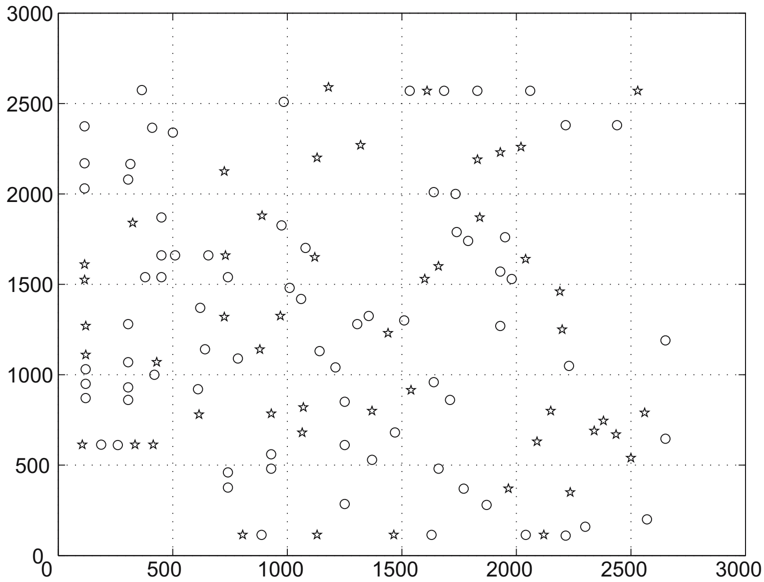
Key Applications

There are many possible queries for a particular set of GIS data. For example, a very basic query for a set of spatiotemporal data would be, "What is the value of interest at a specific location and time instance?" With good interpolation results and efficient representation of the interpolation results in constraint databases, many spatiotemporal queries can be easily answered by query languages. In the following, some examples of Datalog queries are shown.

Ozone Data Example

Based on the ozone data example in Sect. 2, Key Applications, Entry "Constraint Databases and Data Interpolation", some sample spatiotemporal queries are given below. Assume that the input constraint relations are [8]

- *Ozone_orig(x,y,t,w)*, which records the original measured ozone value w at monitoring site location *(x,y)* and time t;
- *Ozone_interp(x,y,t,w)*, which stores the interpolation results of the ozone data by any spatiotemporal interpolation method, such as 3-D shape function or IDW;
- *Ozone_loocv(x,y,t,w)*, which stores the interpolated ozone concentration level at each monitoring site *(x,y)* and time t after applying the leave-one-out cross-validation.



Constraint Database Queries, Figure 1 76 sample houses (◯) and 50 test houses (☆)

Note

The leave-one-out cross-validation is a process that removes one of the n observation points and uses the remaining $n-1$ points to estimate its value; and this process is repeated at each observation point [1]. The observation points are the points with measured original values. For the experimental ozone data, the observation points are the spatiotemporal points *(x,y,t)*, where *(x,y)* is the location of a monitoring site and *t* is the year when the ozone measurement was taken. After the leave-one-out cross-validation, each of the observation points will not only have its original value, but also will have an interpolated value. The original and interpolated values at each observation point can be compared for the purpose of an error analysis. The interpolation error at each data point by calculating the difference between its original and interpolated values is as follows:

$$E_i = \frac{| I_i - O_i |}{O_i} \tag{1}$$

where E_i is the interpolation error at observation point i, I_i is the interpolated value at point i, and O_i is the original value at point i.

Query 2 *For a given location with longitude x and latitude y, find the ozone concentration level in year t.*

This can be expressed in Datalog as follows:

$$Ozone\_value(w) : -Ozone\_interp(x, y, t, w) .$$

Query 3 *Suppose that in the future years, there will be a budget increase so that new ozone monitoring sites can be added. Find the best areas where new monitoring sites should be installed.*

In order to decide the best locations to add new monitoring sites, it is necessary to first find those monitoring sites that have average large interpolation errors according to equation (1), for example, over 20%. Then, do a *buffer* operation on the set of monitoring sites with big errors to find out the areas within certain distance to each site, for example, 50 miles. Since the buffered areas are the areas with poor interpolation result, these areas can be considered the possible areas where new monitoring sites should be built. To find the monitoring sites with more than 20% interpolation errors, perform the following Datalog queries:

$$\begin{aligned} Error(x, y, t, r) : &-Ozone\_orig(x, y, t, w1) , \\ &Ozone\_loocv(x, y, t, w2) , \\ &r = | w1 - w2 | / w1 . \\ Avg\_error(x, y, avg(r)) : &-Error(x, y, t, r) . \\ Sites\_Chosen(x, y) : &-Avg\_error(x, y, ae) , \\ &ae >= 0.2 . \end{aligned}$$

To find the areas within 50 miles to the sites with more than 20% interpolation errors, a GIS *Buffer* operation on the relation *Sites_Chosen* should be performed. The buffer operation is provided by many GIS software packages and the MLPQ constraint database system. After performing the buffer operation, an output relation will be created which contains a 50-mile buffer around the locations stored in the *Sites_Chosen* relation.

Similarly, if there will be a budget cut, similar queries to find out and shut down the monitoring sites with small interpolation errors can be designed.

C

House Price Data Example

The house price data consist of a set of real estate data obtained from the Lancaster county assessor's office in Lincoln, Nebraska. House sale histories since 1990 are recorded in the real estate data set and include sale prices and times. In the experiment, 126 residential houses are randomly selected from a quarter of a section of a township, which covers an area of 160 acres. Furthermore, from these 126 houses, 76 houses are randomly selected as sample data, and the remaining 50 houses are used as test data. Figure 1 shows the 76 houses with circles and the 50 remaining houses with stars.

Tables 2 and 3 show instances of these two data sets. Based on the fact that the earliest sale of the houses in this neighborhood is in 1990, the time is encoded in such a way that 1 represents January 1990, 2 represents February 1990, ..., 148 represents April 2002. Note that some houses are sold more than once in the past, so they have more than one tuple in Table 2. For example, the house at the location (888, 115) was sold three times in the past at time 4 and 76 (which represent 4/1990 and 4/1996) [7].

Assume that the input constraint relations are *House*(*x, y, t, p*) and *Built (x,y,t)*. *House(x,y,t,p)* represents the interpolation result of house price data, and *Built (x,y,t)* records the time *t* (in month) when the house at location *(x,y)* was built. The *Built* relation can be usually easily obtained from real estate or city planning agencies.

Constraint Database Queries, Table 2 Sample *(x, y, t, p)*

X	Y	T	P (price/square foot)
888	115	4	56.14
888	115	76	76.02
1630	115	118	86.02
1630	115	123	83.87
...	...	...	...
2240	2380	51	91.87
2650	1190	43	63.27

Constraint Database Queries, Table 3 Test *(x, y, t)*

X	Y	T
115	1525	16
115	1525	58
115	1525	81
115	1610	63
...	...	...
120	1110	30
615	780	59

Query 4 *For each house, find the starting sale price when the house was built.*

This can be expressed as follows:

$$Start(x, y, p) :- Built(x, y, t),\quad House(x, y, t, p) .$$

Query 5 *Suppose it is known that house prices in general decline for some time after the first sale. For each house, find the first month when it become profitable, that is, the first month when its price exceeded its initial sale price.*

This can be expressed as follows:

$$\begin{aligned} not\_Profitable(x, y, t) &:- Built(x, y, t) . \\ not\_Profitable(x, y, t_2) &:- not\_Profitable(x, y, t_1) , \\ &\quad House(x, y, t_2, p_2) , \\ &\quad Start(x, y, p) , \\ &\quad t_2 = t_1 + 1,\ p_2 < qp . \\ Profitable(x, y, t_2) &:- not\_Profitable(x, y, t_1) , \\ &\quad House(x, y, t_2, p_2) , \\ &\quad Start(x, y, p) , \\ &\quad t_2 = t_1 + 1,\ p_2 > p . \end{aligned}$$

Query 6 *How many months did it take for each house to become profitable?*

This translates as:

$$\begin{aligned} Time\_to\_Profit(x, y, t_3) &:- Built(x, y, t_1) , \\ &\quad Profitable(x, y, t_2) , \\ &\quad t_3 = t_2 - t_1 . \end{aligned}$$

All of the above queries could be a part of a more complex data mining or decision support task. For example, a buyer may want to find out which builders tend to build houses that become profitable in a short time or keep their values best.

Future Directions

Interesting directions for the future work could be to continue to design more interesting queries in spatial constraint databases which can be a valuable part of decision support systems.

Cross References

► Constraint Databases and Data Interpolation
► Constraint Databases and Moving Objects
► Constraint Databases, Spatial
► MLPQ Spatial Constraint Database System

Recommended Reading

1. Hjorth, U.: Computer Intensive Statistical Methods, Validation, Model Selection, and Bootstrap. Chapman and Hall/CRC, London, New York (1994)
2. Jaffar, J., Lassez, J.L.: Constraint Logic Programming. In: Proc. 14th ACM Symposium on Principles of Programming Languages, pp. 111–119 (1987)
3. Kanellakis, P.C., Kuper, G.M., Revesz, P.: Constraint Query Languages. In: ACM Symposium on Principles of Database Systems, pp. 299–313 (1990)
4. Kanellakis, P.C., Kuper, G.M., Revesz, P.: Constraint Query Languages. J. Comput. Sys Sci. **1**, 26–52 (1995)
5. Kuper, G.M., Libkin, L., Paredaens, J. (eds.): Constraint Databases. Springer-Verlag, Berlin, Heidelberg (2000)
6. Li, L.: Spatiotemporal Interpolation Methods in GIS. Ph.D Thesis, University of Nebraska-Lincoln, Lincoln, Nebraska May 2003
7. Li, L., Revesz, P.: A Comparison of Spatio-temporal Interpolation Methods. In: Proc. of the Second International Conference on GIScience 2002. Lecture Notes in Computer Science, vol. 2478, pp. 145–160. Springer, Berlin, Heidelberg, New York (2002)
8. Li, L., Zhang, X., Piltner, R.: A Spatiotemporal Database for Ozone in the Conterminous US, Proc. of the Thirteenth International Symposium on Temporal Representation and Reasoning, pp. 168–176. IEEE (2006)
9. Ramakrishnan, R.: Database Management Systems. McGraw-Hill, New York (1998)
10. Revesz, P.: Introduction to Constraint Databases. Springer, New York (2002)
11. Silberschatz, A., Korth, H., Sudarshan, S.: Database System Concepts, 5th edn. McGraw-Hill, New York (2006)
12. Ullman, J.D.: Principles of Database and Knowledge-Base Systems. Computer Science Press, New York (1989)

Constraint Database Visualization

► Visualizing Constraint Data

Constraint Databases and Data Interpolation

Lixin Li
Department of Computer Sciences,
Georgia Southern University, Statesboro, GA, USA

Synonyms

Spatial interpolation; Spatio-temporal interpolation; Data approximation; Contraint relations; Inverse distance weighting; Splines; Trend surfaces; Fourier series; Shape function; Delaunay triangulation; Nearest neighbors

Definition

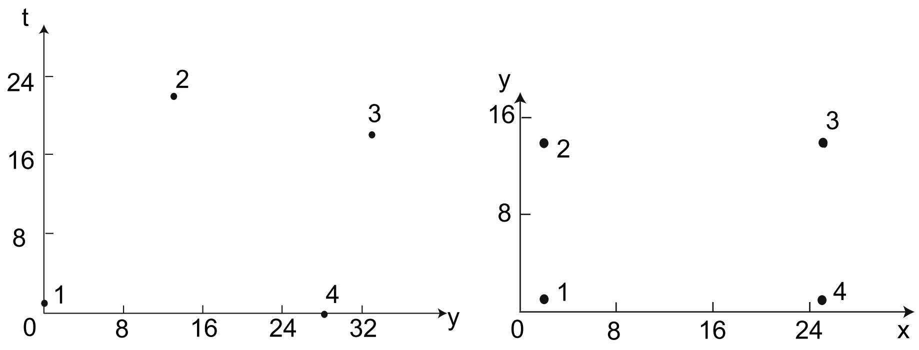
Constraint databases generalize relational databases by finitely representing infinite relations. In the constraint data model, each attribute is associated with an attribute variable and the value of an attribute in a relation is specified implicitly using constraints. Compared with the traditional relational databases, constraint databases offer an extra layer of data abstraction, which is called the *constraint level* [9]. It is the constraint level that makes it possible for computers to use finite number of tuples to represent infinite number of tuples at the logical level.

It is very common in GIS that sample measurements are taken only at a set of points. Interpolation is based on the assumption that things that are close to one another are more alike than those that are farther apart. In order to estimate the values at unsampled points, interpolation is needed.

Constraint databases are very suitable for representing spatial/spatiotemporal interpolation results. In this entry, several spatial and spatiotemporal interpolation methods are discussed, and the representation of their spatiotemporal interpolation results in constraint databases are illustrated by some examples. The performance analysis and comparison of different interpolation methods in GIS applications can be found in [5,6,7].

Historical Background

There exist a number of spatial interpolation algorithms, such as *inverse distance weighting (IDW)*, *kriging*, *splines*, *trend surfaces* and *Fourier series*. Spatiotemporal interpolation is a growing research area. With the additional *time* attribute, the above traditional spatial interpolation algorithms are insufficient for spatiotemporal data and new spatiotemporal interpolation methods must be developed. There have been some papers addressing the issue of spatiotemporal interpolation in GIS. References [1,2,13] deal with the use of spatiotemporal interpolations for different applications. References [4,6] discuss several newly developed *shape function* based spatial/spatiotemporal interpolation methods. There have been some applications on the shape function based methods. For example, reference [7] applies a shape function interpolation method to a set of ozone data in the conterminous US, and reference [6] compares shape function, IDW and kriging based spatiotemporal interpolation methods by using an actual real estate data set with house prices. Reference [13] also uses a shape function based interpolation method to represent the West Nile Virus data in constraint databases and implements a particular epidemiological system called WeNiVIS that enables the visual tracking of and reasoning about the spread of the West Nile Virus epidemic in Pennsylvania.



Constraint Databases and Data Interpolation, Figure 1 The spatial sample points for *Incoming* (*left*) and *Filter* (*right*)

Constraint Databases and Data Interpolation, Table 1 Relational Incoming (*y*, *t*, *u*)

ID	Y	T	U
1	0	1	60
2	13	22	20
3	33	18	70
4	29	0	40

Constraint Databases and Data Interpolation, Table 2 Relational Filter (*x*, *y*, *r*)

ID	X	Y	R
1	2	1	0.9
2	2	14	0.5
3	25	14	0.3
4	25	1	0.8

Scientific Fundamentals

Suppose that the following two sets of sensory data are available in the database [11]:

- *Incoming* (y, t, u) records the amount of incoming ultraviolet radiation u for each pair of latitude degree y and time t, where time is measured in days.
- *Filter* (x, y, r) records the ratio r of ultraviolet radiation that is usually filtered out by the atmosphere above location (x, y) before reaching the earth.

Suppose that Fig. 1 shows the locations of the (y, t) and (x, y) pairs where the measurements for u and r, respectively, are recorded. Then Tables 1 and 2 could be instances of these two relations in a relational database.

The above relational database can be translated into a constraint database with the two constraint relations shown in Tables 3 and 4.

Constraint Databases and Data Interpolation, Table 3 Constrain Incoming (*y*, *t*, *u*)

ID	Y	T	U	
id	y	t	u	$id = 1, y = 0, t = 1, u = 60$
id	y	t	u	$id = 2, y = 13, t = 22, u = 20$
id	y	t	u	$id = 3, y = 33, t = 18, u = 70$
id	y	t	u	$id = 4, y = 29, t = 0, u = 40$

Constraint Databases and Data Interpolation, Table 4 Constraint Filter (*x*, *y*, *r*)

ID	X	Y	R	
id	x	y	r	$id = 1, x = 2, y = 1, r = 0.9$
id	x	y	r	$id = 2, x = 2, y = 14, r = 0.5$
id	x	y	r	$id = 3, x = 25, y = 14, r = 0.3$
id	x	y	r	$id = 4, x = 25, y = 1, r = 0.8$

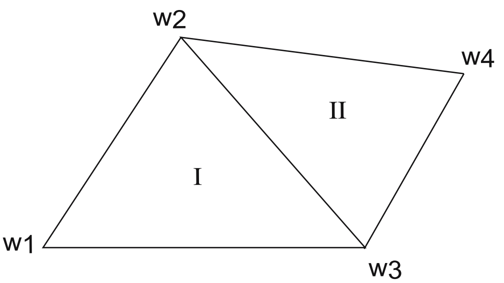
Although any relational relation can be translated into a constraint relation as above, not all the constraint relations can be converted back to relational databases. This is because a constraint relation can store infinite number of solutions. For example, the infinite number of interpolation results of u and r for all the points in the domains for *Incoming* (y,t,u) and *Filter* (x,y,r) can be represented in a constraint database by a finite number of tuples. The representation of interpolation results in constraint databases by different methods for *Incoming* and *Filter* will be given in Key Applications.

Key Applications

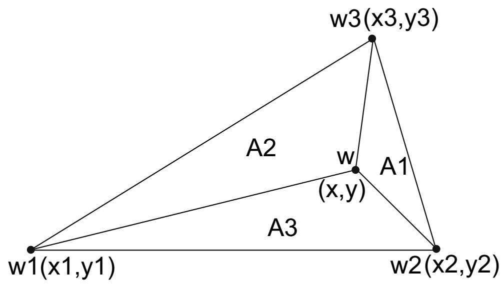
Applications Based on Shape Function Spatial Interpolation

Shape functions, which can be viewed as a spatial interpolation method, are popular in engineering applications, for example, in Finite Element algorithms [15]. There are various types of 2-D and 3-D shape functions. 2-D shape functions for triangles and 3-D shape functions for tetrahedra are of special interest, both of which are linear approximation methods. Shape functions are recently found to be a good interpolation method for GIS applications, and the interpolation results are very suitable to be represented in linear constraint databases [4,5,6,7,11].

2-D Shape Function for Triangles When dealing with complex two-dimensional geometric domains, it is conve-



Constraint Databases and Data Interpolation, Figure 2 Linear Interpolation in Space for Triangular Elements



Constraint Databases and Data Interpolation, Figure 3 Computing shape functions by area divisions

nient to divide the total domain into a finite number of simple sub-domains which can have triangular or quadrilateral shapes. Mesh generation using triangular or quadrilateral domains is important in Finite Element discretization of engineering problems. For the generation of triangular meshes, quite successful algorithms have been developed. A popular method for the generation of triangular meshes is the "Delaunay Triangulation" [8].

A linear interpolation function for a triangular area can be written in terms of three shape functions N_1, N_2, N_3, and the corner values w_1, w_2, w_3. In Fig. 2, two triangular finite elements, I and II, are combined to cover the whole domain considered [6].

In this example, the function in the whole domain is interpolated using four discrete values w_1, w_2, w_3, and w_4 at four locations. A particular feature of the chosen interpolation method is that the function values inside the sub-domain I can be obtained by using only the three corner values w_1, w_2 and w_3, whereas all function values for the sub-domain II can be constructed using the corner values w_2, w_3, and w_4. Suppose $\mathcal{A}$ is the area of the triangular element I. The linear interpolation function for element I can be written as

$$\begin{aligned} w(x,y) &= N_1(x,y)w_1 + N_2(x,y)w_2 + N_3(x,y)w_3 \\ &= [N_1\ N_2\ N_3] \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \end{aligned} \tag{1}$$

where N_1, N_2 and N_3 are the following shape functions:

$$\begin{aligned} N_1(x,y) &= \frac{[(x_2y_3 - x_3y_2) + x(y_2 - y_3) + y(x_3 - x_2)]}{2\mathcal{A}} \\ N_2(x,y) &= \frac{[(x_3y_1 - x_1y_3) + x(y_3 - y_1) + y(x_1 - x_3)]}{2\mathcal{A}} \\ N_3(x,y) &= \frac{[(x_1y_2 - x_2y_1) + x(y_1 - y_2) + y(x_2 - x_1)]}{2\mathcal{A}} \end{aligned} \tag{2}$$

It should be noted that for every sub-domain, a local interpolation function similar to expression (1) is used. Each local interpolation function is constrained to the local triangular sub-domain. For example, the function w of expression (1) is valid only for sub-domain I. For sub-domain II, the local approximation takes a similar form as the expression (1) with replacing the corner values w_1, w_2 and w_3 with the new values w_2, w_3 and w_4.

Alternatively, considering only sub-domain I, the 2-D shape function (2) can also be expressed as follows [11]:

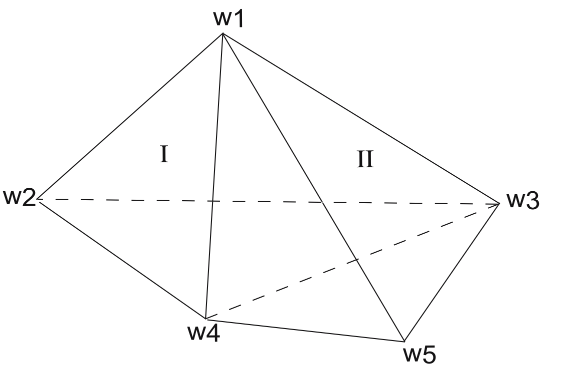
$$N_1(x,y) = \frac{\mathcal{A}_1}{\mathcal{A}},\quad N_2(x,y) = \frac{\mathcal{A}_2}{\mathcal{A}},\quad N_3(x,y) = \frac{\mathcal{A}_3}{\mathcal{A}} \tag{3}$$

where $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ are the three sub-triangle areas of sub-domain I as shown in Fig. 3, and $\mathcal{A}$ is the area of the outside triangle $w_1w_2w_3$.

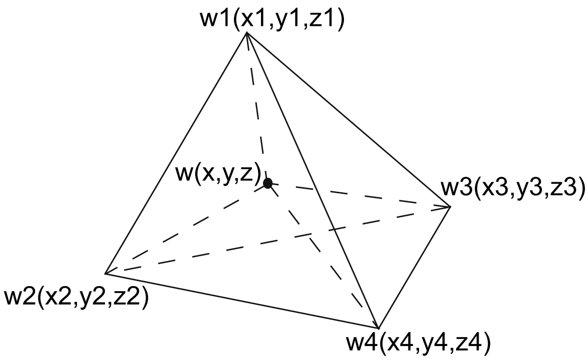
3-D Shape Function for Tetrahedra Three-dimensional domains can also be divided into a finite number of simple sub-domains, such as tetrahedral or hexahedral sub-domains. Tetrahedral meshing is of particular interest. With a large number of tetrahedral elements, complicated 3-D objects can be approximated. There exist several methods to generate automatic tetrahedral meshes, such as the 3-D Delaunay tetrahedrilization and some tetrahedral mesh improvement methods to avoid poorly-shaped tetrahedra.

A linear interpolation function for a 3-D tetrahedral element can be written in terms of four shape functions N_1, N_2, N_3, N_4 and the corner values w_1, w_2, w_3, w_4. In Fig. 4, two tetrahedral elements, I and II, cover the whole domain considered [6].

In this example, the function in the whole domain is interpolated using five discrete values w_1, w_2, w_3, w_4, and w_5 at five locations in space. To obtain the function values inside the tetrahedral element I, the four corner values w_1, w_2, w_3 and w_4 can be used. Similarly, all function values for element II can be constructed using the corner values w_1, w_3, w_4 and w_5. Suppose $\mathcal{V}$ is the volume of the tetrahedral ele-



Constraint Databases and Data Interpolation, Figure 4 Linear Interpolation in Space for Tetrahedral Elements



Constraint Databases and Data Interpolation, Figure 5 Computing shape functions by volume divisions

ment I. The linear interpolation function for element I can be written as:

$$\begin{aligned} w(x, y, z) &= N_1(x, y, z)w_1 + N_2(x, y, z)w_2 \\ &\quad + N_3(x, y, z)w_3 + N_4(x, y, z)w_4 \\ &= [N_1\ N_2\ N_3\ N_4] \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} \end{aligned} \tag{4}$$

where N_1, N_2, N_3 and N_4 are the following shape functions:

$$\begin{aligned} N_1(x, y, z) &= \frac{a_1 + b_1x + c_1y + d_1z}{6\mathcal{V}}, \\ N_2(x, y, z) &= \frac{a_2 + b_2x + c_2y + d_2z}{6\mathcal{V}}, \\ N_3(x, y, z) &= \frac{a_3 + b_3x + c_3y + d_3z}{6\mathcal{V}}, \\ N_4(x, y, z) &= \frac{a_4 + b_4x + c_4y + d_4z}{6\mathcal{V}}. \end{aligned} \tag{5}$$

By expanding the other relevant determinants into their cofactors, there exists

$$a_1 = \det \begin{bmatrix} x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \end{bmatrix} b_1 = -\det \begin{bmatrix} 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \\ 1 & y_4 & z_4 \end{bmatrix}$$

$$c_1 = -\det \begin{bmatrix} x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \\ x_4 & 1 & z_4 \end{bmatrix} d_1 = -\det \begin{bmatrix} x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \\ x_4 & y_4 & 1 \end{bmatrix}$$

with the other constants defined by cyclic interchange of the subscripts in the order 4, 1, 2, 3 [15].

Alternatively, considering only the tetrahedral element I, the 3-D shape function (5) can also be expressed as follows [6]:

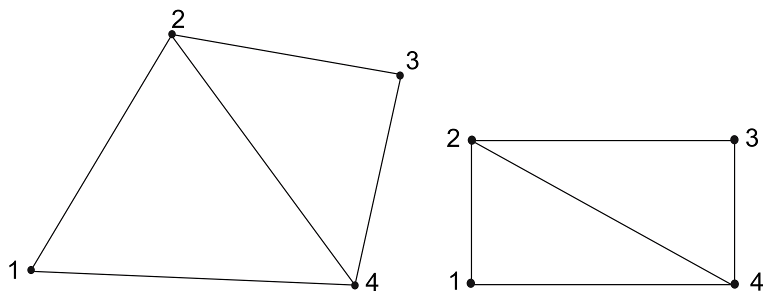
$$\begin{aligned} N_1(x, y, z) &= \frac{\mathcal{V}_1}{\mathcal{V}}, \quad N_2(x, y, z) = \frac{\mathcal{V}_2}{\mathcal{V}}, \\ N_3(x, y, z) &= \frac{\mathcal{V}_3}{\mathcal{V}}, \quad N_4(x, y, z) = \frac{\mathcal{V}_4}{\mathcal{V}}. \end{aligned} \tag{6}$$

$\mathcal{V}_1$, $\mathcal{V}_2$, $\mathcal{V}_3$ and $\mathcal{V}_4$ are the volumes of the four sub-tetrahedra $ww_2w_3w_4$, $w_1ww_3w_4$, $w_1w_2ww_4$, and $w_1w_2w_3w$, respectively, as shown in Fig. 5; and $\mathcal{V}$ is the volume of the outside tetrahedron $w_1w_2w_3w_4$.

Representing Interpolation Results in Constraint Databases In traditional GIS, spatial data are represented in the relational data model, which is the most popular data model. Many database systems are based on the relational model, such as Oracle and MySQL. However, the relational model has disadvantages for some applications, which may lead to infinite relational databases [9]. An infinite relational database means the database has relations with infinite number of tuples. In reality, only a finite set of the tuples can be stored in a relation. Therefore, a finite set of tuples has to be extracted, which leads to data incompleteness. Using constraint databases can solve this infinity problem.

The sensory data of the ultraviolet radiation example in Scientific Fundamentals will be used to illustrate how to represent 2-D shape function spatial interpolation results in constraint databases. In this example, *Incoming*(y, t, u) is treated as if it contains a set of 2-D spatial data. Let *INCOMING*(y, t, u) be the constraint relation that represents the shape function interpolation result of the *Incoming* relation. Similarly, let *FILTER*(x,y,r) be the constraint relation that represents the shape function interpolation result of the *Filter* relation.

Triangulation of the set of sampled points is the first step to use 2-D shape functions. Figure 6 shows the Delaunay



Constraint Databases and Data Interpolation, Figure 6 Delaunay triangulations for *Incoming* (*left*) and *Filter* (*right*)

Constraint Databases and Data Interpolation, Table 5 FILTER (x, y, r) using 2-D shape functions

X	Y	R	
x	y	r	$13x - 23y + 296 \geq 0$, $x \geq 2$, $y \geq 1$, $r = 0.0004x - 0.0031y + 0.1168$
x	y	r	$13x - 23y + 296 < q0$, $x < q25$, $y < q14$, $r = 0.0013x - 0.0038y + 0.1056$

triangulations for the sample points in $Incoming(y, t, u)$ and $Filter(x, y, r)$ illustrated in Fig. 1.

The domain of a triangle can be represented by a conjunction C of three linear inequalities corresponding to the three sides of the triangle. Then, by the shape function (2), the value w of any point x, y inside a triangle can be represented by the following linear constraint tuple:

$$\begin{aligned} R(x, y, w) :- C\,, \quad w = & \\ & [((y_2 - y_3)w_1 + (y_3 - y_1)w_2 + (y_1 - y_2)w_3)/(2\mathcal{A})]x \\ & + [((x_3 - x_2)w_1 + (x_1 - x_3)w_2 + (x_2 - x_1)w_3)/(2\mathcal{A})]y \\ & + [((x_2y_3 - x_3y_2)w_1 + (x_3y_1 - x_1y_3)w_2 \\ & + (x_1y_2 - x_2y_1)w_3)/(2\mathcal{A})]\,. \end{aligned}$$

where $\mathcal{A}$ is a constant for the area value of the triangle. By representing the interpolation in each triangle by a constraint tuple, a constraint relation to represent the interpolation in the whole domain can be found in linear time.

Table 5 illustrates the constraint representation for the interpolation result of *FILTER* using 2-D shape functions. The result of *INCOMING* is similar and the details can be found in reference [11].

Applications Based on Shape Function Spatiotemporal Interpolation

There are two fundamentally different ways for spatiotemporal interpolation: reduction and extension [5]. These methods can be described briefly as follows:

Reduction This approach reduces the spatiotemporal interpolation problem to a regular spatial interpolation case. First, interpolate (using any 1-D interpolation in time) the measured value over time at each sample point. Then get spatiotemporal interpolation results by substituting the desired time instant into some regular spatial interpolation functions.

Extension This approach deals with time as another dimension in space and extends the spatiotemporal interpolation problem into a one-higher dimensional spatial interpolation problem.

Reduction Approach This approach for 2-D space and 1-D time problems can be described by two steps: 2-D spatial interpolation by shape functions for triangles and approximation in space and time. The second step, interpolation in space and time, can be implemented by combining a time shape function with the space approximation function (1).

Assume the value at node i at time t_1 is w_{i1}, and at time t_2 the value is w_{i2}. The value at the node i at any time between t_1 and t_2 can be interpolated using a 1-D time shape function in the following way:

$$w_i(t) = \frac{t_2 - t}{t_2 - t_1} w_{i1} + \frac{t - t_1}{t_2 - t_1} w_{i2}\,. \tag{7}$$

Using the example shown in Fig. 2 and utilizing formulas (1) and (7), the interpolation function for any point constraint to element I at any time between t_1 and t_2 can be expressed as follows [6]:

$$\begin{aligned} w(x, y, t) = & N_1(x, y)\left[\frac{t_2 - t}{t_2 - t_1} w_{11} + \frac{t - t_1}{t_2 - t_1} w_{12}\right] \\ & + N_2(x, y)\left[\frac{t_2 - t}{t_2 - t_1} w_{21} + \frac{t - t_1}{t_2 - t_1} w_{22}\right] \\ & + N_3(x, y)\left[\frac{t_2 - t}{t_2 - t_1} w_{31} + \frac{t - t_1}{t_2 - t_1} w_{32}\right] \\ = & \frac{t_2 - t}{t_2 - t_1} \\ & \cdot \left[N_1(x, y)w_{11} + N_2(x, y)w_{21} + N_3(x, y)w_{31}\right] \\ & + \frac{t - t_1}{t_2 - t_1} \\ & \cdot \left[N_1(x, y)w_{12} + N_2(x, y)w_{22} + N_3(x, y)w_{32}\right]. \end{aligned}$$

The reduction approach for 3-D space and 1-D time problems can be developed in a similar way by combining the 3-D interpolation formula (4) and the 1-D shape function (7). Using the example shown in Fig. 4, the interpolation function for any point constraint to the sub-domain I at any time between t_1 and t_2 can be expressed as follows [6]:

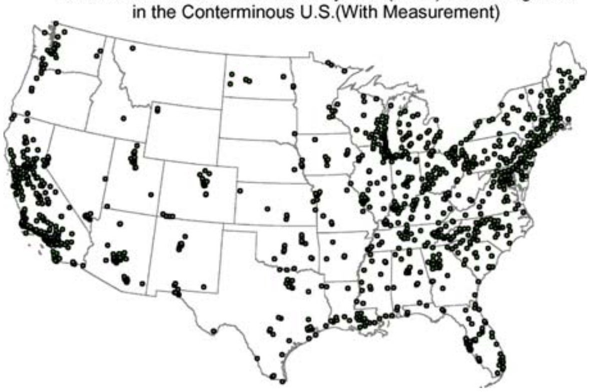
$$\begin{aligned} w(x,y,z,t) = {} & N_1(x,y,z)\left[\frac{t_2-t}{t_2-t_1}w_{11}+\frac{t-t_1}{t_2-t_1}w_{12}\right] \\ & + N_2(x,y,z)\left[\frac{t_2-t}{t_2-t_1}w_{21}+\frac{t-t_1}{t_2-t_1}w_{22}\right] \\ & + N_3(x,y,z)\left[\frac{t_2-t}{t_2-t_1}w_{31}+\frac{t-t_1}{t_2-t_1}w_{32}\right] \\ & + N_4(x,y,z)\left[\frac{t_2-t}{t_2-t_1}w_{41}+\frac{t-t_1}{t_2-t_1}w_{42}\right] \\ = {} & \frac{t_2-t}{t_2-t_1}[N_1(x,y,z)w_{11}+N_2(x,y,z)w_{21} \\ & + N_3(x,y,z)w_{31}+N_4(x,y,z)w_{41}] \\ & + \frac{t-t_1}{t_2-t_1}[N_1(x,y,z)w_{12}+N_2(x,y,z)w_{22} \\ & + N_3(x,y,z)w_{32}+N_4(x,y,z)w_{42}] \,. \end{aligned} \tag{8}$$

Since the 2-D/3-D space shape functions and the 1-D time shape function are linear, the spatiotemporal interpolation function (8) is not linear, but quadratic.

Extension Approach For 2-D space and 1-D time problems, this method treats *time* as a regular third dimension. Since it extends 2-D problems to 3-D problems, this method is very similar to the linear approximation by 3-D shape functions for tetrahedra. The only modification is to substitute the variable z in Eqs. (4)–(6) by the time variable t.

For 3-D space and 1-D time problems, this method treats *time* as a regular fourth dimension. New linear 4-D shape functions based on 4-D Delaunay tesselation can be developed to solve this problem. See reference [3] for details on the 4-D shape functions.

Representing Interpolation Results in Constraint Databases The previous section pointed out the infinity problem for relational databases to represent spatial data. The relational data model shows more disadvantages when handling spatiotemporal data. For example, using the relational model, the current contents of a database (database instance) is a snapshot of the data at a given instant in time. When representing spatiotemporal data, frequent updates have to be performed in order to keep the database instance up to date, which erases the previous database instance. Therefore, the information in the past will be lost. This irrecoverable problem makes the relational data model impractical for handling spatiotemporal data. Using constraint data model can solve this problem. A set of AIRS (Aerometric Information Retrieval System) data will be used to illustrate how spatiotemporal interpolation data can be represented accurately and efficiently in constraint databases.



Constraint Databases and Data Interpolation, Figure 7 1209 AIRS monitoring sites with measurements in the conterminous US

The experimental AIRS data is a set of data with annual ozone concentration measurements in the conterminous US (website www.epa.gov/airmarkets/cmap/data/category1.html). AIRS is a computer-based repository of information about airborne pollution in the United States and various World Health Organization (WHO) member countries. The system is administered by the US Environmental Protection Agency (EPA). The data coverage contains point locations of the monitoring sites for which AIRS data are collected, the annual concentration level measurements of Ozone (O3), and the years of the measurement. Several datasets from the US EPA (website http://cfpub.epa.gov/gdm) were obtained and reorganized into a dataset with schema (x, y, t, w), where x and y attributes are the longitude and latitude coordinates of monitoring site locations, t is the year of the ozone measurement, and w is the O34MAX (4th Max of 1-hr Values for O3) value of the ozone measurement. The original dataset has many zero entries for ozone values, which means no measurements available at a particular site. After filtering out all the zero entries from the original dataset, there are 1209 sites left with measurements. Figure 7 shows the locations of the 1209 monitoring sites [7].

Among the 1209 monitoring sites with measurements, some sites have complete measurements of yearly ozone values from 1994 to 1999, while the other sites have only partial records. For example, some sites only have measurements of ozone values in 1998 and 1999. In total, there

C

are 6135 ozone value measurements recorded. Each measurement corresponds to the ozone value at a spatiotemporal point (x, y, t), where (x, y) is the location of one of the 1209 monitoring sites, and t is a year between 1994 and 1999.

The spatiotemporal interpolation extension method based on 3-D shape functions is implemented into a Matlab program and applied to the AIRS ozone data. The Matlab function *delaunayn* is used to compute the tetrahedral mesh with the 6135 spatiotemporal points as corner vertices. There are 30897 tetrahedra in the resulting mesh. Using the mesh and the original 6135 original ozone values measured at its corner vertices, the annual ozone value at any location and year can be interpolated, as long as the spatiotemporal point is located inside the domain of the tetrahedral mesh.

Since the 3-D shape function based spatiotemporal interpolation Eq. (4) is linear, the interpolation results can be stored in a linear constraint database. Suppose the constraint relation *Ozone_interp* is used to store the interpolation results. Table 6 shows one sample tuple of *Ozone_interp*. The other omitted tuples are of similar format. Since there are 30897 tetrahedra generated in the tetrahedral mesh, there should be 30897 tuples in *Ozone_interp*.

The tuple shown in Table 6 corresponds to the interpolation results of all the points located in the tetrahedron with corner vertices (-68.709, 45.217, 1996), (-68.672, 44.736, 1999), (-67.594, 44.534, 1995), and (-69.214, 45.164, 1999). The ozone values measured at these four points are 0.063, 0.087, 0.096, and 0.074, respectively. In this constraint tuple, there are 10 constraints. The relationship among these constraints is AND. The first four constraints define the four facets of the tetrahedron, the next five constraints give the volume values, and the last constraint is the interpolation function.

Applications Based on IDW Spatial Interpolation

Inverse Distance Weighting (IDW) interpolation [14] assumes that each measured point has a local influence that diminishes with distance. Thus, points in the near neighborhood are given high weights, whereas points at a far distance are given small weights. Reference [12] uses IDW to visualize spatial interpolation data.

The general formula of IDW interpolation for 2-D problems is the following:

$$w(x, y) = \sum_{i=1}^{N} \lambda_i w_i \qquad \lambda_i = \frac{(\frac{1}{d_i})^p}{\sum_{k=1}^{N} (\frac{1}{d_k})^p} \tag{9}$$

where $w(x, y)$ is the predicted value at location (x, y), N is the number of nearest known points surrounding (x, y), λ_i are the weights assigned to each known point value w_i at location (x_i, y_i), d_i are the 2-D Euclidean distances between each (x_i, y_i) and (x, y), and p is the exponent, which influences the weighting of w_i on w.

For 3-D problems, the IDW interpolation function is similar as formula (9), by measuring 3-D Euclidean distances for d_i.

Constraint Databases and Data Interpolation, Table 6 The constraint relation *Ozone_interp*(x, y, t, w), which stores the 3-D shape function interpolation results of the ozone data

X	Y	R	W	
				$0.002532x + 0.003385y + 0.000511t \geq 1$,
				$0.002709x + 0.003430y + 0.000517t \geq 1$,
				$0.002659x + 0.003593y + 0.000511t < q1$,
				$0.002507x + 0.003175y + 0.000515t < q1$,
x	y	t	w	$v = 0.0127$,
				$v_1 = 1/6 \mid 1.71x + 2.17y + 0.35t - 682.87 \mid$,
				$v_2 = 1/6 \mid 2.10x + 2.84y + 0.40t - 790.39 \mid$,
				$v_3 = 1/6 \mid 1.28x + 1.63y + 0.24t - 474.05 \mid$,
				$v_4 = 1/6 \mid 2.53x + 3.38y + 0.51t - 999.13 \mid$,
				$wv = 0.063v_1 + 0.087v_2 + 0.096v_3 + 0.074v_4$
x	y	t	w	⋮
				⋮

Representing Interpolation Results in Constraint Databases To represent the IDW interpolation, the nearest neighbors for a given point should be found. The idea of higher-order Voronoi diagrams (or kth order Voronoi diagrams) can be borrowed from computational geometry to help finding the nearest neighbors. Higher-order Voronoi diagrams generalize ordinary Voronoi diagrams by dealing with k closest points. The ordinary Voronoi diagram of a finite set S of points in the plane is a partition of the plane so that each region of the partition is the locus of points which are closer to one member of S than to any other member [8]. The higher-order Voronoi diagram of a finite set S of points in the plane is a partition of the plane into regions such that points in each region have the same closest members of S. As in an ordinary Voronoi diagram, each Voronoi region is still convex in a higher-order Voronoi diagram. From the definition of higher-order Voronoi diagrams, it is obvious to see that the problem of finding the k closest neighbors for a given point in the whole domain, which is closely related to the IDW interpolation method with $N = k$, is equivalent to constructing kth order Voronoi diagrams.

Although higher-order Voronoi diagrams are very difficult to create by imperative languages, such as C, C++, and Java, they can be easily constructed by declarative languages, such as Datalog. For example, a second-order Voronoi region for points (x_1, y_1), (x_2, y_2) can be expressed in Datalog as follows.

At first, let $P(x, y)$ be a relation that stores all the points in the whole domain. Also let $Dist(x, y, x_1, y_1, d_1)$ be a Euclidean distance relation where d_1 is the distance between (x, y) and (x_1, y_1). It can be expressed in Datalog as:

$$Dist(x, y, x_1, y_1, d_1) :- d_1 = \sqrt{(x - x_1)^2 + (y - y_1)^2}\,.$$

Note that any point (x, y) in the plane does *not* belong to the 2nd order Voronoi region of the sample points (x_1, y_1) and (x_2, y_2) if there exists another sample point (x_3, y_3) such that (x, y) is closer to (x_3, y_3) than to either (x_1, y_1) or (x_2, y_2). Using this idea, the complement can be expressed as follows:

$$\begin{aligned} Not\_2Vor(x, y, x_1, y_1, x_2, y_2) :- \; & P(x_3, y_3),\\ & Dist(x, y, x_1, y_1, d_1),\\ & Dist(x, y, x_3, y_3, d_3),\\ & d_1 > d_3.\end{aligned}$$

$$\begin{aligned} Not\_2Vor(x, y, x_1, y_1, x_2, y_2) :- \; & P(x_3, y_3),\\ & Dist(x, y, x_2, y_2, d_2),\\ & Dist(x, y, x_3, y_3, d_3),\\ & d_2 > d_3.\end{aligned}$$

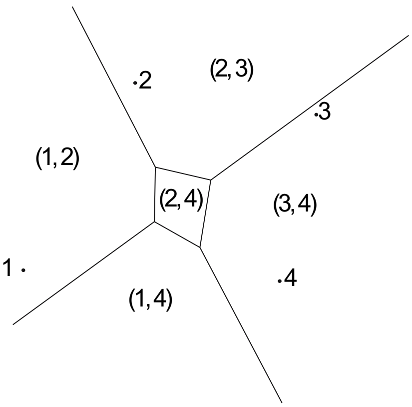
Finally, the negation of the above can be taken to get the 2nd order Voronoi region as follows:

$$\begin{aligned} & 2Vor(x, y, x_1, y_1, x_2, y_2)\\ & :- not\ Not\_2Vor(x, y, x_1, y_1, x_2, y_2)\,. \end{aligned} \tag{10}$$

The second-order Voronoi diagram will be the union of all the nonempty second-order Voronoi regions. Similarly to the 2nd order, any kth-order Voronoi diagram can be constructed.

After finding the closest neighbors for each point by constructing higher-order Voronoi diagrams, IDW interpolation in constraint databases can be represented. The representation can be obtained by constructing the appropriate Nth-order Voronoi diagram and using formula (9).

Based on formula (10), assume that the second-order Voronoi region for points (x_1, y_1), (x_2, y_2) is stored by the relation $Vor_2nd(x, y, x_1, y_1, x_2, y_2)$, which is a conjunction C of some linear inequalities corresponding to the edges of the Voronoi region. Then, using IDW interpolation with $N = 2$ and $p = 2$, the value w of any point (x, y) inside the Voronoi region can be expressed by the constraint tuple as



Constraint Databases and Data Interpolation, Figure 8 The 2nd order Voronoi diagram for *Incoming*

follows:

$$\begin{aligned} R(x, y, w) :- \big(&(x - x_2)^2 + (y - y_2)^2\\ &+ (x - x_1)^2 + (y - y_1)^2\big)w\\ &= \big((x - x_2)^2 + (y - y_2)^2\big)w_1\\ &+ \big((x - x_1)^2 + (y - y_1)^2\big)w_2,\\ &Vor_2nd(x, y, x_1, y_1, x_2, y_2). \end{aligned} \tag{11}$$

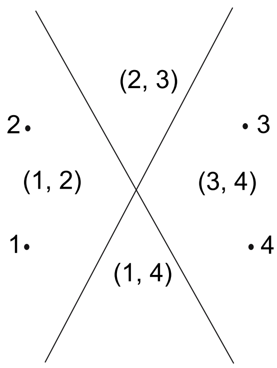
or equivalently as,

$$\begin{aligned} R(x, y, w) :- \big(&(x - x_2)^2 + (y - y_2)^2\\ &+ (x - x_1)^2 + (y - y_1)^2\big)w\\ &= \big((x - x_2)^2 + (y - y_2)^2\big)w_1\\ &+ \big((x - x_1)^2 + (y - y_1)^2\big)w_2,\\ &C. \end{aligned} \tag{12}$$

In the above polynomial constraint relation, there are three variables x, y, and w. The highest order terms in the relation are $2x^2w$ and $2y^2w$, which are both cubic. Therefore, this is a cubic constraint tuple.

The sensory data of the ultraviolet radiation example in Scientific Fundamentals will be used to illustrate how to represent IDW spatial interpolation results in constraint databases. Figures 8 and 9 show the second-order Voronoi diagrams for the sample points in *Incoming(y, t, u)* and *Filter(x, y, r)*, respectively. Please note that some second-order Voronoi regions are empty. For example, there is no (1,3) region in Fig. 8, and there are no (1,3) and (2,4) regions in Fig. 9 [10].

Let *INCOMING(y, t, u)* and *FILTER(x, y, r)* be the constraint relations that store the IDW interpolation results of *Incoming(y, t, u)* and *Filter(x, y, r)*. Based on formula (12), Table 7 shows the result of *FILTER*. Note that the four tuples in Table 7 represent the four second-order Voronoi



Constraint Databases and Data Interpolation, Figure 9 The 2nd order Voronoi diagram for *Filter*

Constraint Databases and Data Interpolation, Table 7 FILTER (x, y, r) using IDW

X	Y	R	
x	y	r	$2x - y - 20 < q0, 12x + 7y - 216 < q0,$
			$((x-2)^2 + (y-14)^2)0.9 + ((x-2)^2 + (y-1)^2)0.5$
			$= (2(x-2)^2 + (y-14)^2 + (y-1)^2)r$
x	y	r	$2x - y - 20 < q0, 12x + 7y - 216 < q0,$
			$((x-25)^2 + (y-1)^2)0.9 + ((x-2)^2 + (y-1)^2)0.8$
			$= (2(y-1)^2 + (x-25)^2 + (x-2)^2)r$
x	y	r	$2x - y - 20 \geq 0, 12x + 7y - 216 \geq 0,$
			$((x-25)^2 + (y-14)^2)0.8 + ((x-25)^2 + (y-1)^2)0.3$
			$= (2(x-25)^2 + (y-14)^2 + (y-1)^2)r$
x	y	r	$2x - y - 20 < q0, 12x + 7y - 216 \geq 0,$
			$((x-25)^2 + (y-14)^2)0.5 + ((x-2)^2 + (y-14)^2)0.3$
			$= (2(y-14)^2 + (x-25)^2 + (x-2)^2)r$

regions in Fig. 9. The result of *INCOMING* is similar and the details can be found in reference [3].

Applications Based on IDW Spatiotemporal Interpolation

Similar as shape functions, IDW is originally a spatial interpolation method and it can be extended by reduction and extension approaches to solve spatiotemporal interpolation problems [3].

Reduction Approach This approach first finds the nearest neighbors of for each unsampled point and calculates the corresponding weights λ_i. Then, it calculates for each neighbor the value at time t by some time interpolation method. If 1-D shape function interpolation in time is used, the time interpolation will be similar to (7). The formula for this approach can be expressed as:

$$w(x, y, t) = \sum_{i=1}^{N} \lambda_i w_i(t) \ , \qquad \lambda_i = \frac{(\frac{1}{d_i})^p}{\sum_{k=1}^{N} (\frac{1}{d_k})^p} \qquad (13)$$

where $d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2}$ and $w_i(t) = \frac{t_{i2} - t}{t_{i2} - t_{i1}} w_{i1} + \frac{t - t_{i1}}{t_{i2} - t_{i1}} w_{i2}$.

Extension Approach Since this method treats time as a third dimension, the IDW based spatiotemporal formula is of the form of (9) with

$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (t_i - t)^2} \, .$$

Future Directions

Interesting directions for the future work could be to represent more interpolation methods in spatial constraint databases, apply more interesting data sets to the interpolation methods, compare the performances of the methods, and animation/visualize the interpolation results.

Cross References

- ► Constraint Database Queries
- ► Constraint Databases and Moving Objects
- ► Constraint Databases, Spatial
- ► Voronoi Diagram

Recommended Reading

1. Gao, J. Revesz, P.: Voting prediction using new spatiotemporal interpolation methods. In: Proc. of the Seventh Annual International Conference on Digital Government Research, San Diego (2006)
2. Li, J., Narayanan, R., Revesz, P.: A shape-based approach to change detection and information mining in remote sensing. In: Chen, C.H. (ed.), Front. Remote Sensing Inf. Process. 63–86. WSP (2003)
3. Li, L.: Spatiotemporal Interpolation Methods in GIS. Ph.D thesis, University of Nebraska-Lincoln, Lincoln, Nebraska, May (2003)
4. Li, L., Li, Y., Piltner, R.: A new shape function based spatiotemporal interpolation method. In: Proc. of the First International Symposium on Constraint Databases 2004, Lecture Notes in Computer Science, vol. 3074, 25–39. Springer, Berlin, Heidelberg, New York (2004)
5. Li, L., Revesz, P.: A comparison of spatio-temporal interpolation methods. In: Proc. of the Second International Conference on GIScience 2002, Lecture Notes in Computer Science, vol. 2478, 145–160. Springer, Berlin, Heidelberg, New York (2002)
6. Li, L., Revesz, P.: Interpolation methods for spatio-temporal geographic data. J. Comput. Environ. Urban Syst. **28**(3):201–227 (2004)
7. Li, L., Zhang, X., Piltner, R.: A spatiotemporal database for ozone in the conterminous u.s. In: Proc. of the Thirteenth International Symposium on Temporal Representation and Reasoning, pp. 168–176. IEEE (2006)
8. Preparata, F.P., Shamos, M.I.: Computational Geometry: An Introduction. Springer-Verlag, Berlin, Heidelberg, New York (1985)
9. Revesz, P.: Introduction to Constraint Databases. Springer, New York (2002)
10. Revesz, P., Li, L.: Constraint-based visualization of spatial interpolation data. In: Proc. of the Sixth International Conference on

Information Visualization, pp. 563–569, IEEE Press, London, England (2002)
11. Revesz, P., Li, L.: Representation and querying of interpolation data in constraint databases. In: Proc. of the Second National Conference on Digital Government Research, pp. 225–228, Los Angeles, California (2002)
12. Revesz, P., Li, L.: Constraint-based visualization of spatiotemporal databases. In: Advances in Geometric Modeling, chapter 20, pp. 263–276. John Wiley, England (2003)
13. Revesz, P., Wu, S.: Spatiotemporal reasoning about epidemiological data. Artificial Intelligence in Medicine, to appear (2006)
14. Shepard, D.: A two-dimensional interpolation function for irregularly spaced data. In: 23nd National Conference ACM, pp. 517–524. ACM (1968)
15. Zienkiewics, O.C., Taylor, R.L.:Finite Element Method, Vol. 1, The Basis. Butterworth Heinemann, London (2000)

Constraint Databases and Moving Objects

LIXIN LI
Department of Computer Sciences,
Georgia Southern University, Statesboro, GA, USA

Synonyms

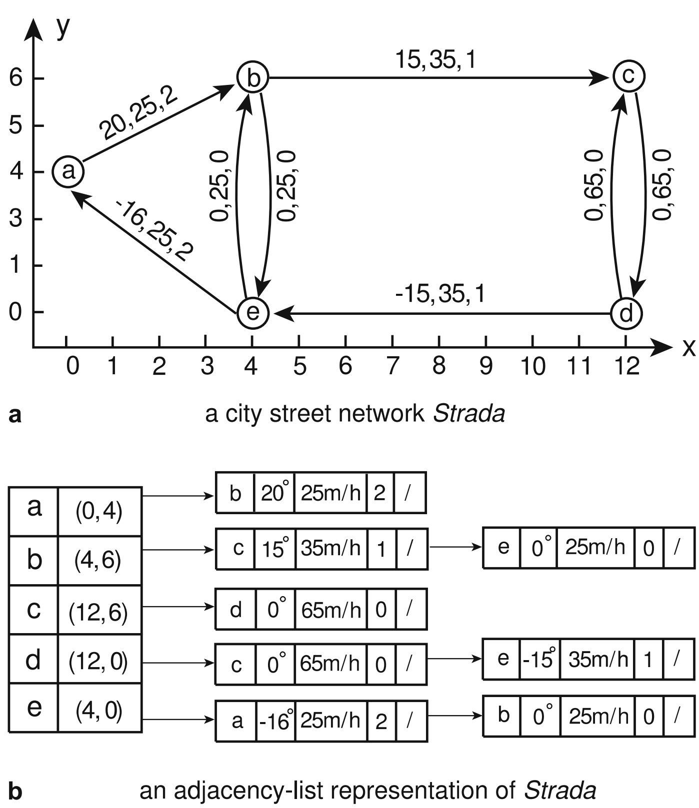
Moving points; Moving regions; Continuously changing maps; Spatio-temporal objects; Moving object constraint databases

Definition

Moving objects can be represented in spatial constraint databases, given the trajectory of the objects.

Historical Background

In general, there are two fundamental ways to abstract moving objects: *moving points* and *moving regions* [3]. Moving points can describe objects for which only the time-dependent position is of interest, while moving regions are able to describe those for which both time-dependent position and spatial extent are of interest. Parametric Rectangles (PReSTO) [8] belong to moving region abstraction. They use growing or shrinking parametric rectangles to model spatiotemporal objects in constraint databases. One advantage in using moving regions is the ability to represent spatial dimensions of objects. Moving points can also be a proper abstraction for moving objects, such as people, animals, stars, cars, planes, ships, and missiles [2]. In reference [10], Saglio and Moreira argued that moving points are possibly the simplest class of continuously changing spatial objects and there are many systems, including those dealing with the position of cars, ships, or planes, which only need to keep the position of the objects. Continuously changing maps are special cases of moving regions. There are many applications that need to be visu-



Constraint Databases and Moving Objects, Figure 1 A city street network *Strada* and its adjacency-list representation

```
Snow_removal(from,to,x,y,t,priority) :- from = "a", to = "b", x =16.8t, y = 4+8.4t,
                                        t >= 0, t <= 14, priority = 2.
Snow_removal(from,to,x,y,t,priority) :- from = "b", to = "c", x = 4+19.3t, y = 6,
                                        t >= 0, t <= 25, priority = 1.
Snow_removal(from,to,x,y,t,priority) :- from = "b", to = "e", x = 4, y = 6-20t,
                                        t >= 0, t <= 18, priority = 0.
Snow_removal(from,to,x,y,t,priority) :- from = "c", to = "d", x = 12, y = 6-20t,
                                        t >= 0, t <= 18, priority = 0.
Snow_removal(from,to,x,y,t,priority) :- from = "d", to = "c", x = 12, y = 20t,
                                        t >= 0, t <= 18, priority = 0.
Snow_removal(from,to,x,y,t,priority) :- from = "d", to = "e", x = 12-20.7t, y = 0,
                                        t >= 0, t <= 23, priority = 1.
Snow_removal(from,to,x,y,t,priority) :- from = "e", to = "a", x = 4-14.7t, y = 14.7t,
                                        t >= 0, t <= 16, priority = 2.
Snow_removal(from,to,x,y,t,priority) :- from = "e", to = "b", x = 4, y = 20t,
                                        t >= 0, t <= 18, priority = 0.
```

Constraint Databases and Moving Objects, Figure 2 Constraint databases representation of the snow removal vehicles for *Strada*

alized by continuously changing maps which will be illustrated in Key Applications.

Scientific Fundamentals

The following example describes how to illustrate the movements of snow removal vehicles in a city street network by moving points in spatial constraint databases.

Suppose that snow removal vehicles are going to clear the snow on the streets of a city. Adjacency-list representation [1] of directed weighted graphs can be applied to model such networks. The city street network *Strada* is shown in Fig. 1a and its adjacency-list representation list is shown in Fig. 1b. Each street has the following attributes: slope, speed limit, and snow clearance priority (the less the value, the higher the priority). These three attributes are shown as labels of each edge in Fig. 1a. They are also displayed in the property fields of each node in Fig. 1b. For example, for the street segment $\overrightarrow{s_{bc}}$, the slope is 15°, the speed limit is 35 mph, and the clearance priority value is 1. The movements of snow removal vehicles in *Strada* can be represented in Fig. 2 by eight Datalog rules in constraint databases.

Constraint Databases and Moving Objects, Table 1 A point-based spatiotemporal relation

Drought_Point			
x (easting)	y (northing)	year	SPI
−315515.56	2178768.67	1992	0.27
−315515.56	2178768.67	1993	−0.17
⋮	⋮	⋮	⋮

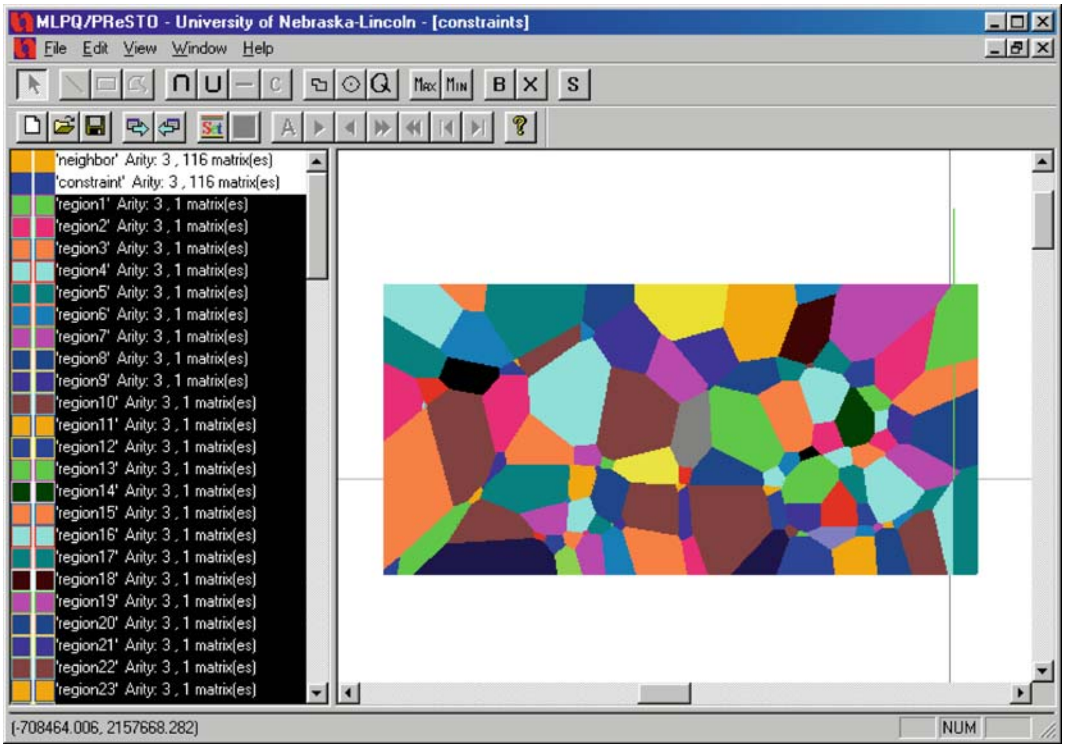
Key Applications

There are many applications that need to be modeled as moving objects, such as continuously changing maps. Constraint databases are capable of handling such applications. The MLPQ (Management of Linear Programming Queries) system is a good example. The MLPQ system is a constraint database system for linear constraint databases [4,7,9]. This system has a graphic user interface (GUI) which supports Datalog-based and icon-based queries as well as visualization and animations. The MLPQ system can outdo the popular ArcGIS system by powerful queries (such as recursive queries) and the ability to dis-

Constraint Databases and Moving Objects, Table 2 A 2nd-order Voronoi region-based database

Drought_Vo2_Space	
$\{(x_1, y_1), (x_2, y_2)\}$	boundary
{ (−9820.18, 1929867.40), (−42164.88, 1915035.54) }	{ (−17122.48, 2203344.58), (3014.51, 2227674.50) (33051.50, 2227674.50), (33051.5, 2140801.51) }
⋮	⋮

Drought_Vo2_Time		
$\{(x_1, y_1), (x_2, y_2)\}$	year	avgSPI
{ (−9820.18, 1929867.4), (−42164.88, 1915035.54) }	1992	−0.47
{ (−9820.18, 1929867.4), (−42164.88, 1915035.54) }	1993	0.71
⋮	⋮	⋮
{ (−507929.66, 2216998.17), (−247864.81, 1946777.44) }	2002	−0.03



Constraint Databases and Moving Objects, Figure 3 The 2rd order Voronoi diagram for 48 weather stations in Nebraska which consists of 116 regions

play continuously changing maps. A few examples are given below.

SPI Spatiotemporal Data

The point-based spatiotemporal relation *Drought_Point (x, y, year, SPI)* stores the average yearly SPI (Standardized Precipitation Index) values sampled by 48 major weather stations in Nebraska from 1992 to 2002. SPI is a common and simple measure of drought which is based solely on the probability of precipitation for a given time period. Values of SPI range from 2.00 and above (extremely wet) to −2.00 and less (extremely dry), with near normal conditions ranging from 0.99 to −0.99. A drought event is defined when the SPI is continuously negative and reaches a value of −1.0 or less, and continues until the SPI becomes positive. The *Drought_Point* relation, as shown in Table 1, was obtained from the Unified Climate Access Network (UCAN) [5].

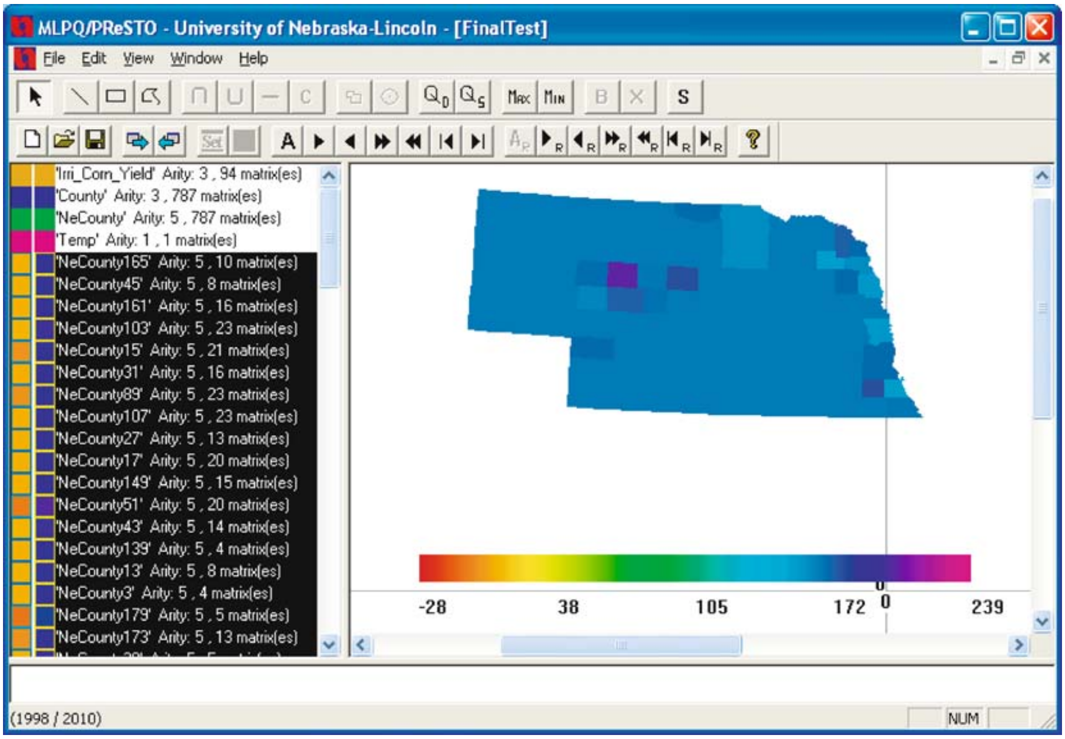
Assume that in the point-based spatiotemporal relation *Drought_Point*, the 48 weather stations have not changed their locations for the last 10 years and measured SPI values every year. The spatial and temporal parts of the 2nd-order Voronoi region-based relation of *Drought_Point* are shown in Table 2.

Continuously changing maps in MLPQ can be used to visualize the 2nd-order Voronoi diagrams. Users need to push the color animation button in the MLPQ GUI and

Constraint Databases and Moving Objects, Table 3 A region-based spatiotemporal database with separate spatial and temporal relations

Nebraska_Corn_Space_Region	
county	boundary
1	{ (−656160.3, 600676.8), (−652484.0, 643920.3), (−607691.1, 639747.6), (−608934.8, 615649.0), (−607875.6, 615485.8), (−610542.0, 576509.1), (−607662.7, 576138.5), (−611226.9, 537468.5), (−607807.7, 536762.1), (−608521.1, 527084.0), (−660885.4, 531441.2), (−661759.8,532153.1) }
⋮	⋮

Nebraska_Corn_Time_Region					
county	year	practice	acres	yield	production
1	1947	irrigated	2700	49	132300
1	1947	non-irrigated	81670	18	1470060
1	1947	total	84370	19	1602360
⋮	⋮	⋮	⋮	⋮	⋮



Constraint Databases and Moving Objects, Figure 4 A snapshot of continuously changing maps for county-based corn yield in Nebraska when $t = 1998$

input the following three parameters: the beginning time instance, ending time instance and step size. Then, the color of each region of the map will be animated according to its value at a specific time instance. Figure 3 shows the 2nd-order Voronoi diagram for the 48 weather stations in Nebraska at the snapshot when $t = 1992$ [5].

NASS Spatiotemporal Data

A NASS (National Agricultural Statistics Service) region-based spatiotemporal database shows the yearly corn yield and production in each county of the state of Nebraska. The spatial part of the database is shown in the upper half of Table 3 which uses the vector representation of counties in Nebraska, while the temporal part is shown in the lower half of Table 3 [6].

Continuously changing maps can be used to animate the total corn yield in each county in Nebraska during a given period. First, each county polygon needs to be represented in MLPQ. Although such county vector data in the US are usually available in ArcView shape file format, a program can be implemented to convert ArcView shape files to MLPQ input text files. The conversion from MLPQ files to shape files can also be implemented. Figure 4 shows the snapshot during the color map animation when $t = 1998$ [5].

Future Directions

Interesting directions for future work could be to continue the discovery of moving objects that are difficult to model in relational databases but can be conveniently modeled in spatial constraint databases, and to extend the MLPQ system so as to improve the visualization/animation power of the system.

Cross References

► Constraint Database Queries
► Constraint Databases and Data Interpolation
► Constraint Databases, Spatial
► MLPQ Spatial Constraint Database System
► Visualization of Spatial Constraint Databases

Recommended Reading

1. Cormen, T.H., Leiserson, C.E., Rivest, R.: Introduction to Algorithms. McGraw-Hill, New York (1999)
2. Erwig, M., Güting, R., Schneider, M.,Vazirgiannis, M.: Spatio-temporal data types: An approach to modeling and querying moving objects in databases. ChoroChronos Research Project, Technical Report CH-97–08 (1997)
3. Güting, R.: An introduction to spatial database systems. VLDB J. **3**(4),357–399 (1994)

4. Kanjamala, P., Revesz, P., Wang, P.: MLPQ/GIS: A GIS using linear constraint databases. In: Prabhu, C.S.R. (ed.): Proc. 9th COMAD International Conference on Management of Data, pp. 389–393. Hyderabad, India (1998)
5. Li, L.: Spatiotemporal Interpolation Methods in GIS. Ph.D Thesis, University of Nebraska-Lincoln, Lincoln, Nebraska, May 2003
6. Li, L., Revesz, P.: The relationship among GIS-oriented spatiotemporal databases. In: Proc. of the Third National Conference on Digital Government Research, Boston (2003)
7. Revesz, P.: Introduction to Constraint Databases. Springer, New York (2002)
8. Revesz, P., Cai, M.: Efficient querying of periodic spatiotemporal objects. Ann. Math. Artif. Intell. **36**(4):437–457 (2002)
9. Revesz, P., Li, Y.: MLPQ: A linear constraint database system with aggregate operators. In: Proc. 1st International Database Engineering and Applications Symposium, pp. 132–137. IEEE Press, Washington, DC (1997)
10. Saglio, J.-M., Moreira, J.: Oporto: A realistic scenario generator for moving objects. In: Proc. DEXA'99 Workshop on Spatiotemporal Data Models and Languages (STDML), pp. 426–432, Florence, Italy (1999)

Constraint Databases, Spatial

PETER REVESZ
Department of Computer Science and Engineering, University of Nebraska, Lincoln, NE, USA

Synonyms

Databases, relational; Query, datalog

Definition

Spatial constraint databases form a generalization of relational databases for the purpose of representing spatial and spatiotemporal data. Whereas in a relational database each table is a finite set of tuples, in a spatial constraint database each table is a finite set of quantifier-free conjunctions of atomic constraints. In spatial constraint databases the most frequent type of atomic constraints used are linear equations and linear inequalities. The variables of the atomic constraints correspond to the attributes in the relation, hence they are called attribute variables.

As an example from [12] consider the highly simplified map of the town of Lincoln, Nebraska shown in Fig. 1.

This map can be represented in a spatial constraint database with linear inequality constraints as follows.

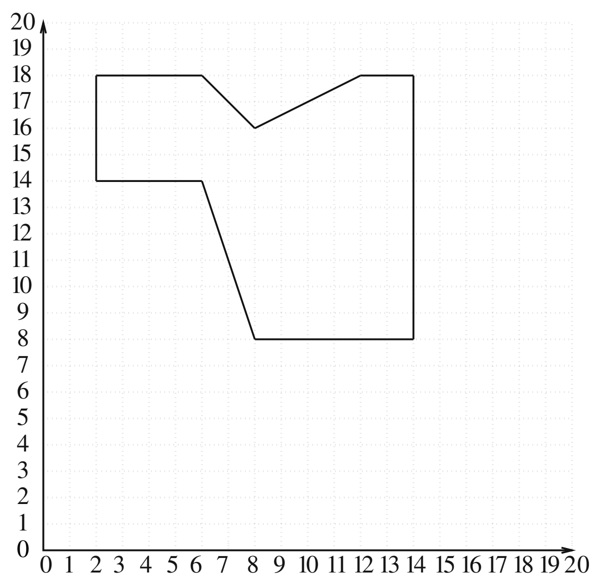
Constraint Databases, Spatial, Table 1 Lincoln

X	Y	
x	y	$y \geq x + 8,\ y \geq 14,\ x \geq 2,\ y < q18,\ y < q - x + 24$
x	y	$y < qx + 8,\ y < qx/2 + 12,\ y < q18,\ x < q14,\ y \geq 8,\ y \geq -3x + 32$

In the above the attribute variables x and y represent the longitude and latitude, respectively, as measured in units from the (0,0) point in the above map. In general, any polygonal shape can be represented by first dividing it into a set of convex polygons and then representing each convex polygon with n sides by a conjunction of n linear inequality constraints. Note that the above town map is a concave polygon, but it can be divided along the line $y = x + 8$ into two convex polygons. The convex pentagon above line $y = x + 8$ is represented by the first row of the constraint table, while the convex hexagon below line $y = x + 8$ is represented by the second row of the constraint table. Within any row, the atomic constraints are connected by commas, which simply mean conjunction. While the atomic constraints can be given in any order in each row, it is customary to present them in an order that corresponds to a clockwise ordering of the sides of the convex polygon that they together represent.

Historical Background

Constraint databases, including spatial constraint databases, were proposed by Kanellakis, Kuper and Revesz in 1990 [7]. A much-delayed journal version of their original conference paper appeared in 1995 [8]. These papers considered a number of constraint database query languages and challenged researchers to investigate further their properties. Benedikt et al. [2] showed that relational calculus queries of constraint databases when the



Constraint Databases, Spatial, Figure 1 A map of Lincoln, Nebraska constraint database contains polynomial constraints over the reals cannot express even simple Datalog-expressible queries. On the other hand, Datalog queries with linear constraints can already express some computationally hard or even undecidable problems. Only in special cases, such as with gap-order constraints of the form $x - y \geq c$ where x and y are integer or rational variables and c is a non-negative constant can an algorithm be given for evaluating Datalog queries [11].

The above results influenced researchers to implement several spatial constraint database systems with non-recursive query languages, usually some variation of non-recursive SQL and linear equality and linear inequality constraints. These systems include, in historical order, the MLPQ system [3], the CCUBE system [3], the DEDALE system [5], and the CQA/CDB system [4]. The MLPQ system implements both SQL and Datalog queries.

Constraint databases are reviewed in a number of books. Chapter 5.6 of [1], a standard reference in database theory, is a compact description of the main ideas of constraint databases. [9] is a collection of research articles devoted to constraint databases. It is a good introduction to already advanced researchers. Revesz [12] is the standard textbook for the subject. It is used at many universities. Chapter 4 of [15], which is an excellent source on all aspects of spatial databases, is devoted exclusively to constraint databases. Chapter 6 of [6], which is a sourcebook on moving object databases, is also devoted exclusively to constraint databases.

Scientific Fundamentals

The semantics of logical models of a spatial constraint database table is a relational database that contains all the rows that can be obtained by substituting values into the attribute variables of any constraint row such that the conjunction of constraints in that row is true. For example, it is easy to see that the semantics of the spatial constraint database table *Lincoln* is a relation that contains all (x, y) points that belong to the town map. Since there are an infinite number of such (x, y) points when x and y are real numbers, spatial constraint databases are also called finitely representable infinite relational databases.

Spatial constraint databases can represent not only areas but also boundaries by using linear equality constraints. Representing the boundary of an n-ary (concave or convex) polygonal area, requires n rows in a constraint table. For example, the boundary of the town of Lincoln, Nebraska can be represented as shown in Table 2.

In the above each range constraint of the form $a < qx < qb$ is an abbreviation of $a < qx,\ x < qb$ where x is any variable and a and b are constants.

Constraint Databases, Spatial, Table 2 Lincoln_Boundary

X	Y		
x	y	$y = 18,$	$2 < qx < q6$
x	y	$y = -x + 24,$	$6 < qx < q8$
x	y	$y = x/2 + 12,$	$8 < qx < q12$
x	y	$y = 18,$	$12 < qx < q14$
x	y	$x = 14,$	$8 < qy < q18$
x	y	$y = 8,$	$8 < qx < q14$
x	y	$y = -3x + 32,$	$6 < qx < q8$
x	y	$y = 14,$	$2 < qx < q6$
x	y	$x = 2,$	$14 < qy < q18$

Spatial constraint databases can be extended to higher dimensions. For example, a Z attribute for height or a T attribute for time can be added. As an example, suppose that Fig. 1 shows the map of Lincoln, Nebraska in year 2000, and since then the town has expanded to the east continuously at the rate of one unit per year. Then the growing town area between years 2000 and 2007 can be represented as shown in Table 3.

Spatial constraint databases with polynomial constraints are also possible. With the increased complexity of constraints, more complex spatial and spatiotemporal objects can be represented. For example, suppose that an airplane flies over Lincoln, Nebraska. Its shadow can be represented as a spatial constraint database relation **Airplane_Shadow** using polynomial constraints over the variables x, y and t. (Here the time unit t will be measured in seconds and not years as in the **Lincoln_Growing** example.)

Spatial constraint databases can be queried by the same query languages that relational databases can be queried. For example, the popular Structured Query Language (SQL) for relational databases is also applicable to spatial constraint databases. For example, the following query finds when the towns of Lincoln, Nebraska and Omaha, Nebraska will grown into each other.

```
SELECT Min(Lincoln_Growing.T)
  FROM Lincoln_Growing, Omaha_Growing
 WHERE Lincoln_Growing.X = Omaha_Growing.X
       AND
       Lincoln_Growing.Y = Omaha_Growing.Y
       AND
       Lincoln_Growing.T = Omaha_Growing.T
```

Suppose that the airplane flies over Lincoln, Nebraska. The next query finds when the shadow of the airplane will leave the town.

```
  SELECT Max(Airplane_Shadow.T)
    FROM Lincoln, Airplane_Shawdow
   WHERE Lincoln.X = Airplane_Shadow.X AND
         Lincoln.Y = Airplane_Shadow.Y
```

Besides SQL queries, spatial constraint databases can be queried by relational calculus queries (that is, first-order logic queries) and by Datalog queries.

Key Applications

Many applications of spatial constraint database systems are similar to the applications of other GIS systems, such as the ARC/GIS system. These applications typically include problems where various kinds of maps, road networks, and land utilization information are represented and overlaying of different maps plays a key role in information processing and querying. However, efficiently describing and querying spatial or geographic data is just one application area of spatial constraint databases.
Spatial constraint databases are also useful in applications that go beyond traditional GIS systems. These applications typically require the representation of moving objects or spatiotemporal data and high-level, often recursive, SQL or Datalog queries. The following are some examples of such applications.

Applications Based on Interpolated Spatiotemporal Data

Spatiotemporal data, just like spatial data, often contain missing pieces of information that need to be estimated based on the known data. For example, given the prices of the houses when they were sold during the past 30 years in a town, one may need to estimate the prices of those houses which were not sold at any time within the past 30 years. In such applications the results of the interpolation can be represented as a spatial constraint database with x, y fields for the location of houses, t field for time, and a p field for price [10]. The value of p will be estimated as some function, often a linear equation, of the other variables. If the spatiotemporal interpolation result is represented in a spatial constraint database like MLPQ, then it becomes easy to to use the data for applications like estimating price and tax payments for particular houses, estimating total taxes received by the town from sale of houses in any subdivision of the town etc.

Constraint Databases, Spatial, Table 3 Lincoln_Growing

X	Y	T	
x	y	t	$y \geq x+8,\ y \geq 14,\ x \geq 2,\ y < q18,$ $y < q - x + 24,\ 2000 < qt < q2007$
x	y	t	$y < qx+8,\ y < qx/2+12,\ y < q18,$ $x < q14+(t-2000),\ y \geq 8,$ $y \geq -3x+32,\ 2000 < qt < q2007$

Applications Based on Continuously Changing Maps

In many applications maps are not static but continuously changing. For example, the regions where drought occurs or where an epidemic occurs change with time within every country or state. Such changing maps are conveniently represented in spatial constraint databases similarly to the representation of the growing town area [14]. In these cases too, when the changing map representations are available in a spatial constraint database, many types of applications and specific queries can be developed. For example, a drought monitoring application can estimate the damage done to insured agricultural areas. Here the changing drought map needs to be overlayed with static maps of insured areas and particular crop areas. Another example is tracking the spread of an epidemics and estimating when it may reach certain areas of the state or country and how many people may be effected.

Applications Based on Moving Objects

Moving objects can be animate living beings, that is, people and animals, natural phenomena such as hurricanes and ocean currents, or man-made moving objects such as airplanes, cars, missiles, robots, and trains. Moving objects can also be represented in spatial constraint databases. The representation can then be used in a wide range of applications from weather prediction to airplane and train scheduling, or some combination application. For example, one may need to find which airplanes are in danger of being influenced by a hurricane. Endangered airplanes can be given a warning and rerouted if need be. Another application is checking whether the airspace surrounding an airport ever gets too crowded by the arriving and departing airplanes.

Future Directions

Spatial constraint database systems that implement polynomial constraints are being developed. There is a growing number of spatial constraint database applications. Improved algorithms for indexing and querying spatial constraint data are also being developed. Finally, improved high-level visualization methods of constraint data are sought after to enhance the user interface of spatial constraint database systems.

Cross References

- ► Constraint Database Queries
- ► Constraint Databases and Data Interpolation
- ► Constraint Databases and Moving Objects
- ► Indexing Spatial Constraint Databases
- ► Linear Versus Polynomial Constraint Databases
- ► MLPQ Spatial Constraint Database System
- ► Visualization of Spatial Constraint Databases

Recommended Reading

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley, Reading, Massachusetts (1995)
2. Benedikt, M., Dong, G., Libkin, L., Wong, L.: Relational expressive power of constraint query languages. J. ACM **45**(1), 1–34 (1998)
3. Brodsky, A., Segal, V., Chen, J., Exarkhopoulo, P.: The CCUBE constraint object-oriented database system. Constraints **2**(3–4), 245–77 (1997)
4. Goldin, D., Kutlu, A., Song, M., Yang, F.: The Constraint Database Framework: Lessons Learned from CQA/CDB. Proc. International Conference on Data Engineering, pp. 735–737 (2003)
5. Grumbach, S., Rigaux, P., Segoufin, L.: The DEDALE system for complex spatial queries. In: Proc. ACM SIGMOD International Conference on Management of Data, pp. 213–24 (1998)
6. Güting, R.H., Schneider, M.: Moving Objects Databases. Morgan Kaufmann, Amsterdam (2005)
7. Kanellakis, P.C., Kuper, G.M., Revesz, P.: Constraint query languages. In: Proc. ACM Symposium on Principles of Database Systems, pp. 299–313 (1990)
8. Kanellakis, P.C., Kuper, G.M., Revesz, P.: Constraint query languages. J. Comput. Syst. Sci. **51**(1), 26–52 (1995)
9. Kuper, G.M., Libkin, L., Paredaens, J. (eds.): Constraint Databases. Springer-Verlag, Berlin (2000)
10. Li, L., Revesz, P.: Interpolation methods for spatio-temporal geographic data. Comput. Environ. Urban Syst. **28**, 201–227 (2004)
11. Revesz, P.: A closed-form evaluation for Datalog queries with integer (gap)-order constraints. Theor. Comput. Sci. **116**(1), 117–49 (1993)
12. Revesz, P.: Introduction to Constraint Databases. Springer, New York (2002)
13. Revesz, P., Li, Y.: MLPQ: A linear constraint database system with aggregate operators. In: Proc. 1st International Database Engineering and Applications Symposium, pp. 132–7. IEEE Press, Washington (1997)
14. Revesz, P., Wu, S.: Spatiotemporal reasoning about epidemiological data. Artificial Intelligence in Medicine, to appear (2006)
15. Rigaux, P., Scholl, M., Voisard, A.: Spatial Databases with Application to GIS. Morgan Kaufmann, San Francisco(2002)

Constraint Query Languages

► Constraint Database Queries

Constraints, Authority

► Time Geography

Constraints, Capability

► Time Geography

Constraints, Coupling

► Time Geography

Content Metadata

► Feature Catalogue

Context-Aware

► User Interfaces and Adaptive Maps

Context-Aware Dynamic Access Control

► Security Models, Geospatial

Context-Aware Presentation

► Information Presentation, Dynamic

Context-Aware Role-Based Access Control

► Security Models, Geospatial

Context-Sensitive Visualization

► Information Presentation, Dynamic

Contextualization

► Geospatial Semantic Web: Personalisation

Contingency Management System

► Emergency Evacuation Plan Maintenance

Continuity Matrix

► Spatial Weights Matrix

Continuity Network

► Conceptual Neighborhood

Continuous Location-Based Queries

► Continuous Queries in Spatio-temporal Databases

Continuous Queries

► Indexing, Query and Velocity-Constrained
► Queries in Spatio-temporal Databases, Time Parameterized

Continuous Queries in Spatio-temporal Databases

Xiaopeng Xiong[1], Mohamed F. Mokbel[2], Walid G. Aref[1]
[1] Department of Computer Science, Purdue University, West Lafayette, IN, USA
[2] Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA

Synonyms

Continuous location-based queries; Long-running spatiotemporal queries; Moving queries; Continuous Query Processing

Definition

A continuous query is a new query type that is issued once and is evaluated continuously in a database server until the query is explicitly terminated. The most important characteristic of continuous queries is that their query result does not only depend on the present data in the databases but also on continuously arriving data. During the execution of a continuous query, the query result is updated continuously when new data arrives. Continuous queries are essential to applications that are interested in transient and frequently updated objects and require monitoring query results continuously. Potential applications of continuous queries include but are not limited to real-time location-aware services, network flow monitoring, online data analysis and sensor networks.

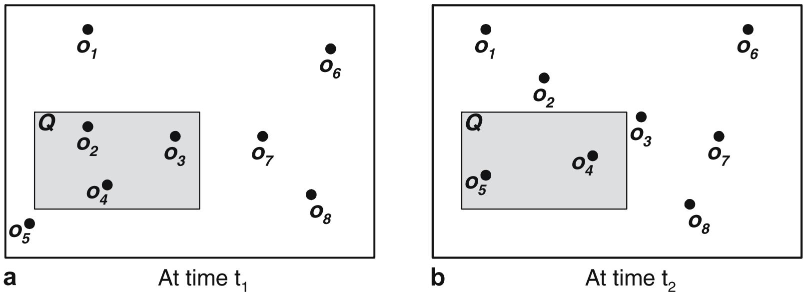
Continuous queries are particularly important in *Spatiotemporal Databases*. Continuous spatio-temporal queries are evaluated continuously against spatiotemporal objects and their results are updated when interested objects change spatial locations or spatial extents over time. Figure 1 gives an example of a continuous query in a spatiotemporal database. In Fig. 1, o_1 to o_8 are objects moving in the data space and Q is a continuous spatiotemporal query that tracks moving objects within the shaded query region. As plotted in Fig. 1a, the query answer of Q with respect to time t_1 consists of three objects: $\{o_2, o_3, o_4\}$. Assume that at a later time t_2, the objects change their locations as shown in Fig. 1b. Particularly, o_2 and o_3 move out of the query region while o_5 moves inside the query region. o_4 also moves, however it remains inside the query region. Due to the continuous evaluation of Q, the query answer of Q will be updated to $\{o_4, o_5\}$ at time t_2.

Historical Background

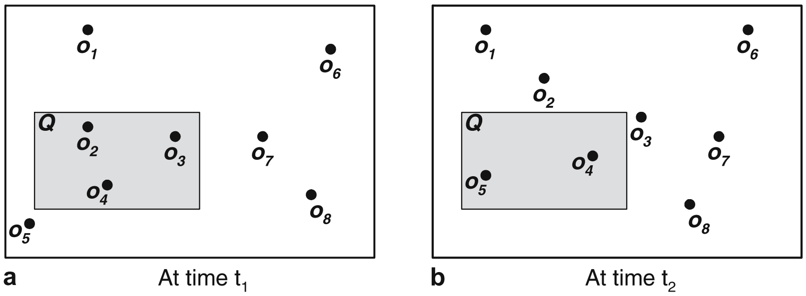
The study of continuous spatiotemporal queries started in the 1990s as an important part of the study of *Spatiotemporal Databases*. Since then, continuous spatiotemporal queries have received increasing attention due to the advances and combination of portable devices and locating technologies. Recently, the study of continuous queries in spatiotemporal databases has become one of the most active fields in the database domain.

Scientific Fundamentals

There are various types of continuous queries that can be supported in spatiotemporal databases. In general, continuous spatiotemporal queries can be classified into various categories based on different classifying criteria. The



Continuous Queries in Spatio-temporal Databases, Figure 1 An example of continuous query



Continuous Queries in Spatio-temporal Databases, Figure 2 Continuous Spatiotemporal Query Types Based on Mobility

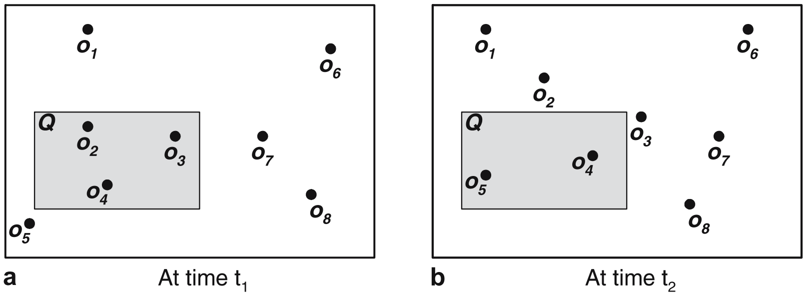
most common classifying criteria are based on the type of query interest, the mobility of query interest and the time of query interest [3].

According to the type of query interest, there are a wide variety of continuous spatiotemporal queries. The following describes some interesting query types that are widely studied.

- *Continuous range queries* [1,4]. This type of continuous query is interested in spatiotemporal objects inside or overlapping with a given spatial region. Continuous range queries have many important applications and are sometimes used as a filter step for other types of continuous queries.
  Examples:
  "Continuously report the number of trucks on Road US-52."
  "Continuously show me all the hotels within 3 miles during my drive."
- *Continuous k-nearest-neighbor (CkNN) queries* [8,10,11]. A CkNN query is a query tracking continuously the k objects that are the nearest ones to a given query point. The objects of interest and/or the query point may move during query evaluation.
  Examples:
  "Continuously track the nearest maintenance truck to my vehicle (in the battlefield)."
  "Continuously show me the 10 nearest hotels during my drive."
- *Continuous Reverse Nearest-neighbor (CRNN) queries* [2,9]. A CRNN query continuously identifies a set of objects that have the querying object as their nearest neighbor object.
  Examples:
  "Continuously find soldiers who need my help (I am the nearest doctor to him/them)."
  "Continuously send electronic advertisement of our hotel to vehicles that have our hotel as their nearest hotel."

Based on the mobility of interest, continuous spatiotemporal queries can be classified as *moving queries over static objects*, *static queries over moving objects* and *moving queries over moving objects* [3].

- *Moving queries over static objects*. In this query category, the objects of interest are static while the query region or query point of the continuous spatiotemporal query may change over time. This query type is abstracted in Fig. 2a. In Fig. 2a, the query o_2 moves along with time (e. g., at t_1, t_2 and t_3) and the objects (represented by black dots) are stationary.
  Examples:
  "Continuously return all the motels within 3 miles to John during John's road-trip."
  "Continuously find the nearest gas station to Alice while she is driving along Highway IN-26."
  In the examples above, the objects of interest (i. e., the gas stations and the motels) are static objects and the query regions/points (i. e., the 3-mile region surrounding John's location and the location of Alice) are continuously changing.
- *Static queries over moving objects*. In this query category, the query region/point of continuous spatiotemporal query remains static while the objects of interest are continuously moving. This query type is abstracted in Fig. 2b. As plotted in Fig. 2b, the objects keep moving along with time (e. g., at t_1, t_2 and t_3) while the query Q is stationary.
  Examples:
  "Continuously monitor the number of buses in the campus of Purdue University."
  "Continuously find the nearest 100 taxis to a certain hotel."
  In the above examples, the query region (i. e., the university campus) or the query point (i. e., the hotel) does not move while the objects of interest (i. e., the buses and the taxis) continuously move.
- *Moving queries over moving objects*. In this query category, both the query region/point of continuous spatiotemporal query and the objects of interest are capable of moving. This query type is abstracted in Fig. 2c. As shown in Fig. 2c, the query Q and the objects are both moving over time (e. g., at t_1, t_2 and t_3).
  Example:
  "Continuously report all the cars within 1 mile when the sheriff drives along the State street."



Continuous Queries in Spatio-temporal Databases, Figure 3 Continuous Spatiotemporal Query Types Based on the Time of Interest

In this example, the query region (i. e., 1-mile region surrounding the location of the sheriff) and the objects of interest (i. e., the cars) are both moving.

Based on the time of query interest, continuous spatiotemporal queries can be classified as *historical queries*, *present queries* and *future queries* [3]. Figure 3 plots the three types of queries. In Fig. 3, the gray dots represent historical object locations and the black dots represent current object locations. The dotted lines with arrows represent anticipated object movements in the future based on the objects' current velocities.

- *Historical queries*. These types of continuous queries are interested in events of the past. Historical queries are especially interesting to applications in data warehouse analysis and business intelligence.
  Example:
  "Continuously calculate the average number of trucks on Highway I-65 for the past 2 hours."
  In the above example, the query result depends on the historical data of the past two hours and is continuously updated when time evolves.
- *Present queries*. In this query category, continuous queries are evaluated against only the current status of spatiotemporal objects. Present queries are important to real-time monitoring and location-based services.
  Examples:
  "Continuously return the total number of vehicles in the monitored area."
  "Send an alarm once the child steps out of the neighbor's home."
  The query results of the examples above depend only on the current locations of the objects of interest (i. e., the locations of vehicles and the location of the child).
- *Future queries*. In this query type, the query results are based on the predication of future events. The evaluation of future queries usually relies on the knowledge of expected object movement, such as the velocity information of the objects. Future queries are particularly useful for alarm systems and risk prevention.
  Example:
  "Send an alarm if two aircrafts are becoming less than 5 miles apart from each other in 3 minutes."
  The above query can be evaluated continuously based on the velocity information of the aircrafts. When the velocities of the aircrafts are changed, the query is re-evaluated and the query result may change accordingly.

Key Applications

Continuous spatiotemporal queries have numerous applications in a wide range of fields.

Traffic Monitoring

Continuous spatiotemporal queries can be applied to monitor real-time traffic conditions in interested areas. Abnormal traffic conditions such as vehicle collision and traffic congestion can be detected and monitored by continuously analyzing incoming traffic data. Besides traffic detection and monitoring, recommendations of alternative routes can be sent to drivers, allowing them to bypass the slow-traffic roads.

Traffic Pattern Detection

Traffic usually demonstrates a repeated pattern with respect to location and time. Detection of such a pattern is important to predict the traffic conditions in the future at the interested area. Continuous spatiotemporal queries can be used to detect such patterns by continuously analyzing traffic data and maintaining spatio-temporal histograms.

Danger Alarming

Depending on the present or predicted locations of interested objects, continuous queries can trigger alarms to prevent potential dangers. Danger alarming has applications in both daily life and national security. For example, alarms can be sent when kids step out of the home backyard or when an unidentified flight is predicted to fly over a military base in five minutes.

Digital Battlefield

In the digital battlefield, continuous queries can help commanders make decisions by continuously monitoring the context of friendly units such as soldiers, tanks and flights.

Road-Trip Assistance

Travel by driving will become more convenient with the aid of continuous spatiotemporal queries. A driver can be continuously informed about information such as nearby hotels, gas stations and grocery stores. More dynamic information such as nearby traffic conditions and weather alarms based on the current location can also be supported by integrating corresponding information in spatiotemporal databases.

Location-Based E-commerce

Continuous spatiotemporal queries can be utilized to shift E-commerce from web-based E-commerce to location-based E-commerce. Location-based E-commerce is E-commerce associated with the locations of potential customers. One example of location-based E-commerce is "sending coupons to all vehicles that are within twenty miles of my hotel".

Climate Analysis and Predicting

Climatology study can benefit from employing continuous spatiotemporal queries over climate data. Climate phenomena such as hurricanes, storms and cumulonimbus clouds can be modeled as spatiotemporal objects with changing spatial extents and moving locations. Climate phenomena can be continuously monitored and it is plausible to analyze historical climate phenomena and to predict future climate phenomena.

Environmental Monitoring

Continuous spatiotemporal queries can work with sensor networks to monitor environmental phenomena such as forest fires or polluted water domains. Sensors in a sensor network continuously detect environmental events and feed the data into spatiotemporal databases. Then, the properties of the environmental phenomena (e. g., the shape and the movement of the fire or the polluted water area) can be continuously monitored.

Future Directions

The study of continuous spatiotemporal queries is steadily progressing. Current efforts focus on a variety of challenging issues [5,6,7]. The following provides some directions among a long list of research topics.

- *Query Language.* This direction is to define an expressive query language so that any continuous spatiotemporal queries can be properly expressed.
- *Novel Query Types.* More query types are proposed based on new properties of spatiotemporal data. Examples of new continuous spatiotemporal query types include continuous probabilistic queries, continuous group nearest queries etc.
- *Query Evaluation.* Due to the continuous evaluation of queries, efficient query evaluation algorithms are needed to process queries incrementally and continuously whenever data is updated.
- *Data Indexing.* Traditional data indexing structures usually do not perform well under frequent object updates. There is a challenging issue on designing update-tolerant data indexing to cope with continuous query evaluation.
- *Scalability.* When the number of moving objects and the number of continuous queries become large, the performance of spatiotemporal databases will degrade and cannot provide a timely response. Increasing the scalability of spatiotemporal databases is an important topic to address.

Cross References

► Queries in Spatio-temporal Databases, Time Parameterized
► Spatio-temporal Database Modeling with an Extended Entity-Relationship Model

Recommended Reading

1. Kalashnikov, D.V., Prabhakar, S., Hambrusch S.E., Aref, W.A.: Efficient evaluation of continuous range queries on moving objects. In: DEXA '02: Proceedings of the 13th International Conference on Database and Expert Systems Applications, pp. 731–740. Springer, Heidelberg (2002)
2. Kang, J.M., Mokbel, M.F., Shekhar, S., Xia, T., Zhang, D.: Continuous evaluation of monochromatic and bichromatic reverse nearest neighbors. In: ICDE (2007)
3. Mokbel, M.F., Aref, W.A., Hambrusch, S.E., Prabhakar, S.: Towards scalable location-aware services: Requirements and research issues. In: GIS (2003)
4. Mokbel, M.F., Xiong, X., Aref, W.A.: SINA: Scalable incremental processing of continuous queries in spatio-temporal databases. In: SIGMOD (2004)
5. Roddick, J.F., Hoel, E., Egenhofer, M.E., Papadias, D., Salzberg, B.: Spatial, temporal and spatio-temporal databases – hot issues and directions for phd research. SIGMOD Rec., **33**(2), 126–131 (2004)
6. Sellis, T.: Chorochronos – research on spatiotemporal database systems. In: DEXA '99: Proceedings of the 10th International Workshop on Database & Expert Systems Applications, p. 452. IEEE Computer Society, Washington, DC, USA (1999)
7. Sellis, T.K.: Research issues in spatio-temporal database systems. In: SSD '99: Proceedings of the 6th International Symposium on Advances in Spatial Databases, pp. 5–11. Springer, London (1999)
8. Tao, Y., Papadias, D., Shen, Q.: Continuous nearest neighbor search. In: VLDB (2002)
9. Xia, T., Zhang, D.: Continuous reverse nearest neighbor monitoring. In: ICDE '06: Proceedings of the 22nd International Confer-

ence on Data Engineering (ICDE '06), p. 77. Washington, DC, USA (2006)
10. Xiong, X.; Mokbel, M.F. Aref, W.G.: SEA-CNN: Scalable processing of continuous K-nearest neighbor queries in spatio-temporal databases. In: ICDE (2005)
11. Li, J.Y.Y.; Han, J.: Continuous k-nearest neighbor search for moving objects. In: SSDBM '04: Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM'04), p. 123. IEEE Computer Society, Washington, DC, USA (2004)

Continuous Query Processing

► Continuous Queries in Spatio-temporal Databases

Continuously Changing Maps

► Constraint Databases and Moving Objects

Contraflow for Evacuation Traffic Management

BRIAN WOLSHON
Department of Civil and Environmental Engineering, Louisiana State University, Baton Rouge, LA, USA

Synonyms

Reversible and convertible lanes; One-way-out evacuation; Emergency preparedness; Evacuation planning; All-lanes-out; Split designs; Merge designs

Definition

Contraflow is a form of reversible traffic operation in which one or more travel lanes of a divided highway are used for the movement of traffic in the opposing direction[1] [1]. It is a highly effective strategy because it can both immediately and significantly increase the directional capacity of a roadway without the time or cost required to plan, design, and construct additional lanes. Since 1999, contraflow has been widely applied to evacuate regions of the southeastern United States (US) when under threat from hurricanes. As a result of its recent demonstrated effectiveness during Hurricane Katrina [2], it also now looked upon as a potential preparedness measure for other mass-scale hazards.

[1]The common definition of contraflow for evacuations has been broadened over the past several years by emergency management officials, the news media, and the public to include the reversal of flow on any roadway during an evacuation.

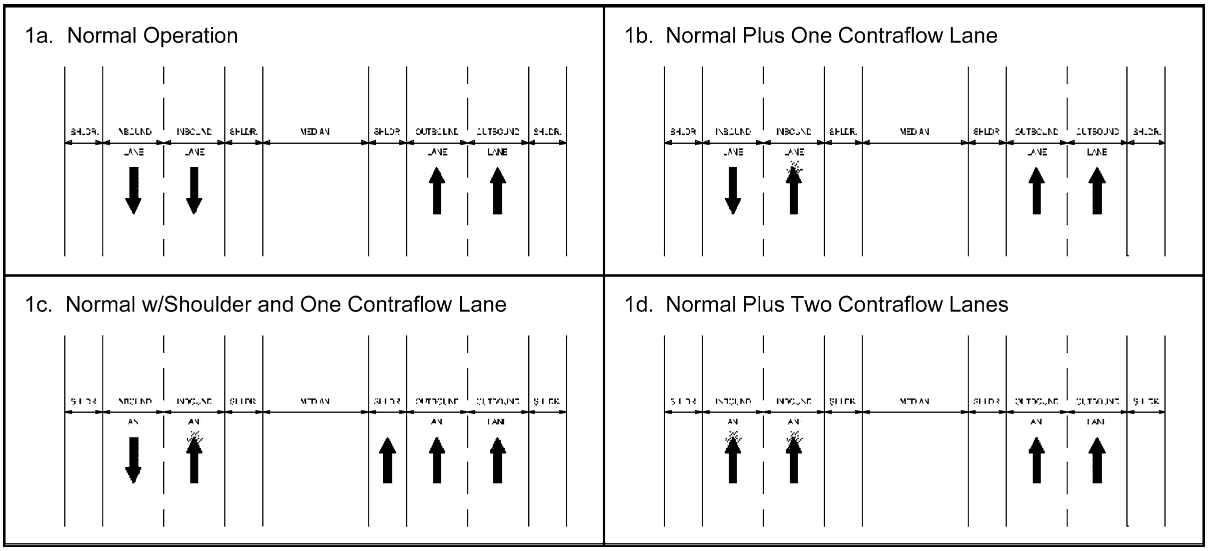
Contraflow segments are most common and logical on freeways because they are the highest capacity roadways and are designed to facilitate high speed operation. Contraflow is also more practical on freeways because these routes do not incorporate at-grade intersections that interrupt flow or permit unrestricted access into the reversed segment. Freeway contraflow can also be implemented and controlled with fewer manpower resources than unrestricted highways.

Nearly all of the contraflow strategies currently planned on US freeways have been designed for the reversal of all inbound lanes. This configuration, shown schematically in Inset 1d of Fig. 1, is commonly referred to as a "One-Way-Out" or "All-Lanes-Out" evacuation. Though not as popular, some contraflow plans also include options for the reversal of only one of the inbound lanes (Inset 1b) with another option to use one or more of the outbound shoulders (Inset 1c) [3]. Inbound lanes in these plans are maintained for entry into the threat area by emergency and service vehicles to provide assistance to evacuees in need along the contraflow segment.

Historical Background

Although evacuation-specific contraflow is a relatively recent development, its application for other types of traffic problems is not new [4]. In fact, various forms of reversible traffic operation have been used throughout the world for decades to address many types of directionally unbalanced traffic conditions. They have been most common around major urban centers where commuter traffic is heavy in one direction while traffic is light in the other. Reverse and contraflow operations have also been popular for managing the infrequent, but periodic and predictable, directionally imbalanced traffic patterns associated with major events like concerts, sporting events, and other public gatherings. Reversible lanes have also been cost effective on bridges and in tunnels where additional directional capacity is needed, but where additional lanes can not be easily added.

While the date of the first use of contraflow for an evacuation is not known with certainty, interest in its potential began to be explored after Hurricane Andrew struck Florida in 1992. By 1998, transportation and emergency management officials in both Florida and Georgia had plans in place to use contraflow on segments of Interstate freeways. Ultimately, the watershed event for evacuation contraflow in the United States was Hurricane Floyd in 1999. Since then, every coastal state threatened by hurricanes has developed and maintains plans for the use of evacuation contraflow.



Contraflow for Evacuation Traffic Management, Figure 1 Freeway contraflow lane use configurations for evacuations [3]

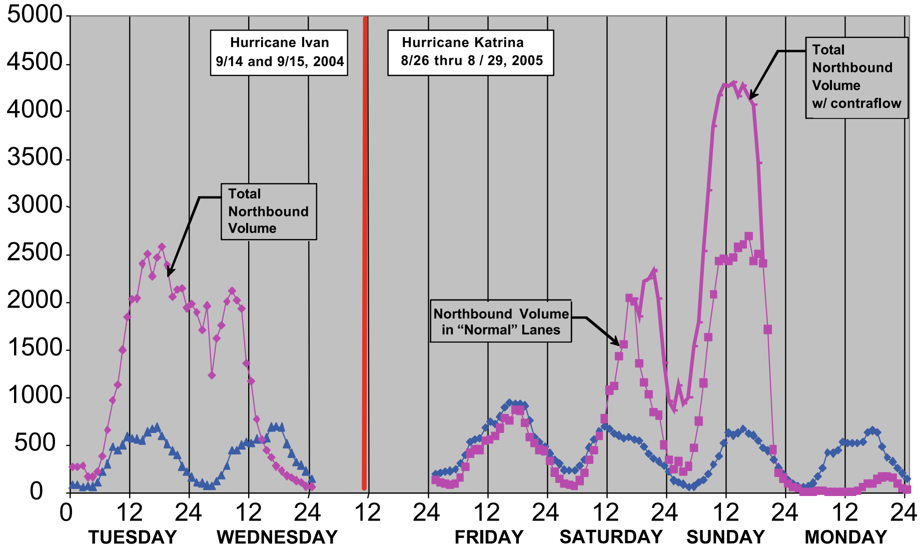
Hurricane Floyd triggered the first two major implementations of contraflow, one on a segment of Interstate (I) 16 from Savannah to Dublin, Georgia and the other on I-26 from Charleston to Columbia, South Carolina. The results of both of these applications were generally positive, although numerous areas for improvement were also identified. The contraflow application in South Carolina was particularly interesting because it was not pre-planned. Rather, it was implemented on an improvisational basis after a strong public outcry came from evacuees trapped for hours in congested lanes of westbound I-26 seeking ways to use the near-empty eastbound lanes.

The first post-Floyd contraflow implementations occurred in Alabama for the evacuation of Mobile and Louisiana for the evacuation New Orleans. Once again, many lessons were learned and numerous improvements in both physical and operational aspects of the plans were suggested. The timing of these events was quite fortuitous for New Orleans. Within three months of the major changes that were implemented to the Louisiana contraflow plan after Hurricane Ivan, they were put into operation for Hurricane Katrina. The changes, so far the most aggressive and far-ranging of any developed until that time [5], involved the closure of lengthy segments of interstate freeway, forced traffic onto alternative routes, established contraflow segments across the state boundary into Mississippi, coordinated parallel non-freeway routes, and reconfigured several interchanges to more effectively load traffic from surface streets. The results of these changes were reflected in a clearance time for the city that was about half of the previous prediction [6].

Scientific Fundamentals

Although the basic concept of contraflow is simple, it can be complex to implement and operate in actual practice. If not carefully designed and managed, contraflow segments also have the potential to be confusing to drivers. To insure safe operation, improper access and egress movements must be prohibited at all times during its operation. Segments must also be fully cleared of opposing traffic prior to initiating contraflow operations. These are not necessarily easy to accomplish, particularly in locations where segments are in excess of 100 miles and where interchanges are frequent. For these reasons some transportation officials regard them to be risky and only for use during daylight hours and under the most dire situations. They are also the reason why contraflow for evacuation has been planned nearly exclusively for freeways, where access and egress can be tightly controlled.

To now, contraflow evacuations have also been used only for hurricane hazards and wildfires and no other type of natural or manmade hazard. The first reason for this is that these two hazards affect much greater geographic areas and tend to be slower moving relative to other hazards. Because of their scope they also create the need move larger numbers of people over greater distances than other types of hazards. The second reason is that contraflow requires considerable manpower and materiel resources as well as time to mobilize and implement. Experiences in Alabama and Louisiana showed that the positioning of traffic control devices and enforcement personnel takes at least six hours not including the time to plan and prepo-



Contraflow for Evacuation Traffic Management, Figure 2
Northbound traffic volume – I-55 at Fluker Louisiana (Data source: LA DOTD)

sition equipment for the event. In Florida, where needs are great and manpower resources are stretched thin, evacuation contraflow requires involvement from the Florida National Guard. For this reason (among others), Florida officials require a minimum of 49 hours of advanced mobilization time for contraflow to be implemented [7].

Operational Effects of Contraflow

As the goal of an evacuation is to move as many people as quickly out of the hazard threat zone as possible, the primary goal of contraflow is to increase the rate of flow and decrease the travel time from evacuation origins and destinations. Prior to field measurement, it was hypothesized that the flow benefits of contraflow would be substantial, but less than that of an equivalent normally flowing lane [3]. These opinions were based on measurements of flow on I-26 during the Hurricane Floyd evacuation and the theory that drivers would drive at slower speeds and with larger spacing in contraflow lanes.

The highest flow rates measured by the South Carolina Department of Transportation (DOT) during the Floyd evacuation were between 1,500 to 1,600 vehicles per hour per lane (vphpl) [8]. Traffic flows measured during the evacuations for Hurricanes Ivan and Katrina on I-55 in Louisiana were somewhat less than the South Carolina rates. Flows in the normal-flow lanes of I-55 averaged about 1,230 vphpl during the peak 10 hours of the evacuation. Flow rates in the contraflow lanes during the same period averaged about 820 vphpl. These volumes compare to daily peaks of about 400 vphpl during routine periods and a theoretical capacity of 1,800 to 2,000 vphpl for this segment.

The graph of Fig. 2 illustrates the hourly traffic flow on I-55 during the evacuations for Hurricanes Ivan (when contraflow was not used) and Katrina (when contraflow was used). During the 48 hour period of the Ivan evacuation (shown on the left side of the graph) a total of 60,721 vehicles traveled northbound through this location. During the Katrina evacuation, the total volume was 84,660 vehicles during a corresponding 48 hour period. It is also worthy to note that the duration of the peak portion of the evacuation (i. e., when the volumes were noticeably above the prior three week average) was about the same for both storms.

The data in Fig. 2 are also of interest because they are consistent with prior analytical models of evacuation that have estimated maximum evacuation flow on freeways with contraflow to be about 5,000 vph. One of the difficulties in making full analyses of evacuation volume in general, and of contraflow volume in specific, has been a lack of speed data. Although the flow rates recorded during the two recent Louisiana hurricane evacuations are considerably below than the theoretical capacity of this section of freeway, it can not be determined with certainty if the conditions were congested with low operating speeds and small headways or relatively free flowing at more moderate levels of demand. It is also interesting to note that empirical observation of speed at a point toward the end of the segment did not appear to support the popular theory of elevated driver caution during contraflow. In fact, traffic enforcement personnel in Mississippi measured speeds well in excess of posted speed limits as the initial group of drivers moved through the newly opened lanes.

Elements of Contraflow Segments

Reversible roadways have a number of physical and operational attributes that common among all applications. The principle physical attributes are related to spatial characteristics of the design, including its overall length, number of lanes, as well as the configuration and length of the inbound and outbound transition areas. The primary operational attributes are associated with the way in which the segment will be used and include the temporal control of traffic movements. The temporal components of all reversible lane segments include the frequency and duration of a particular configuration and the time required to transition traffic from one direction to another. The duration of peak-period commuter reversible applications, for example, typically last about 2 hours (not including set-up, removal, and transition time) with a twice daily frequency. Evacuation contraflow, however, may only be implemented once in several years, its duration of operation may last several days.

Like all reversible flow roadways, contraflow lanes need to achieve and maintain full utilization to be effective. Although this sounds like an obvious fact, it can be challenging to achieve in practice. The most common reason for underutilization has been inadequate transitions into and out of the contraflow segment. Contraflow requires a transition section at the inflow and outflow ends to allow drivers to maneuver into and out of the reversible lanes from the unidirectional lanes on the approach roadways leading into it. Since these termini regulate the ingress and egress of traffic entering and exiting the segment and they are locations of concentrated lane changing as drivers weave and merge into the desired lane of travel, they effectively dictate the capacity of the entire segment.

Through field observation and simulation studies [9,10] it has been shown that contraflow entry points with inadequate inflow transitions result in traffic congestion and delay prior to the contraflow segment and prohibit the segment from carrying capacity-level demand. This was illustrated by I-10 contraflow segment in New Orleans during the Hurricane Ivan evacuation. At that time, evacuating traffic vehicles in the left and center outbound lanes of I-10 were transitioned across the median and into the contraflow lanes using a paved crossover. However, the combination of the crossover design, temporary traffic control devices, presence of enforcement personnel, and weaving vehicles created a flow bottleneck that restricted inflow into the contraflow lanes. This caused two problems. First, it limited the number of vehicles that could enter the contraflow lanes limiting flow beyond the entry point significantly below its vehicle carrying capability. The other was that it caused traffic queues upstream of the crossover that extended back for distances in excess of 14 miles. This plan was significantly improved prior to the Katrina evacuation one year later by permitting vehicles to enter the contraflow lanes at multiple points, spatially spreading the demand over a longer distance and reducing the length and duration amount of the congested conditions [5].

Inadequate designs at the downstream end of contraflow segments can also greatly limit its effectiveness. Prior experience and simulation modeling [11] have shown that an inability to move traffic from contraflow lanes back into normally flowing lanes will result in congestion backing up from the termination transition point in the contraflow lanes. Under demand conditions associated with evacuations, queue formation can occur quite rapidly and extend upstream for many miles within hours. To limit the potential for such scenarios, configurations that require merging of the normal and contraflowing lanes are discouraged; particularly if they also incorporate lane drops. Two popular methods that are used to terminate contraflow include routing the two traffic streams at the termination on to separate routes and reducing the level of outflow demand at the termination by including egress point along the intermediate segment. Several of the more common configurations are discussed in the following section.

Contraflow Plans and Designs

The primary physical characteristics of contraflow segments are the number of lanes and the length. A 2003 study [12] of hurricane evacuation plans revealed that 18 controlled access evacuation contraflow flow segments and three additional arterial reversible roadway segments have been planned for use in the US. Currently, all of the contraflow segments are planned for a full “One-Way-Out” operation. The shortest of the contraflow freeway segments was the I-10 segment out of New Orleans at about 25 miles long. The longest were two 180 segments of I-10 in Florida; one eastbound from Pensacola to Tallahassee and the other westbound from Jacksonville to Tallahassee. Most of the others were between 85 and 120 miles.

In the earliest versions of contraflow, nearly all of the planned segments that were identified in the study were initiated via median crossovers. Now that single point loading strategies have been shown to be less effective, many locations are changing to multi-point loading. Most popular of these are median crossovers, with supplemental loading via nearby reversed interchange ramps.

The termination configurations for the reviewed contraflow segments were broadly classified into one of two groups. The first were *split designs*, in which traffic in the normal and contraflowing lanes were routed onto separate roadways at the terminus. The second group were the *merge designs* in which the separate lane groups are reunited into the normal-flow lanes using various geometric and control schemes. The selection of one or the other of these termination configurations at a particular location by an agency

Contraflow for Evacuation Traffic Management, Table 1 Planned Contraflow/Reverse flow Evacuation Routes [12]

State	Route(s)	Approx. Distance (miles)	Origin Location	Termination Location
New Jersey	NJ-47/ NJ-347* Atlantic City Expressway NJ-72/ NJ-70* NJ-35* NJ-138/I-195	19 44 29.5 3.5 26	Dennis Twp Atlantic City Ship Bottom Boro Mantoloking Boro Wall Twp	Maurice River Twp Washington Twp Southampton Pt. Pleasant Beach Upper Freehold
Maryland	MD-90	11	Ocean City	US 50
Virginia*	I-64	80	Hampton Roads Bridge	Richmond
North Carolina	I-40	90	Wilmington	Benson (I-95)
South Carolina	I-26	95	Charleston	Columbia
Georgia	I-16	120	Savannah	Dublin
Florida	I-10 Westbound I-10 Eastbound SR 528 (Beeline) I-4 Eastbound I-75 Northbound FL Turnpike I-75 (Alligator Alley)	180 180 20 110 85 75 100	Jacksonville Pensacola SR 520 Tampa Charlotte County Ft. Pierce Coast	Tallahassee Tallahassee SR 417 Orange County I-275 Orlando Coast
Alabama	I-65	135	Mobile	Montgomery
Louisiana	I-10 Westbound I-10/I-59 (east/north)	25 115*	New Orleans New Orleans	I-55 Hattiesburg*
Texas	I-37	90	Corpus Christi	San Antonio

(*Notes: Delaware and Virginia contraflow plans are still under development. The actual length of the New Orleans, LA to Hattiesburg, MS contraflow segment will vary based on storm conditions and traffic demand. Since they are undivided highways, operations on NJ-47/NJ-347, NJ-72/NJ-70, and NJ-35 are "reverse flow" rather than contraflow.)

has been a function of several factors, most importantly the level of traffic volume and the configuration and availability of routing options at the end of the segment.

In general, split designs offer higher levels of operational efficiency of the two designs. The obvious benefit of a split is that it reduces the potential for bottleneck congestion resulting from merging four lanes into two. Its most significant drawback is that it requires one of the two lane groups to exit to a different route, thereby eliminating route options at the end of the segment. In some older designs, the contraflow traffic stream was planned to be routed onto an intersecting arterial roadway. One of the needs for this type of split design is adequate capacity on the receiving roadway.

Merge termination designs also have pros and cons. Not surprisingly, however, these costs and benefits are nearly the exact opposite of split designs in their end effect. For example, most merge designs preserve routing options for evacuees because they do not force vehicles on to adjacent roadways and exits. Unfortunately, the negative side to this is that they also have a greater potential to cause congestion since they merge traffic into a lesser number of lanes. At first glance it would appear illogical to merge two high volume roadways into one. However, in most locations where they are planned exit opportunities along the intermediate segment will be maintained to decrease the volumes at the end of the segment.

Key Applications

The list of applications for contraflow continues to grow as transportation and emergency preparedness agencies recognize its benefits. As a result, the number of locations that are contemplating contraflow for evacuations is not known. However, a comprehensive study of contraflow plans [12] in 2003 included 21 reverse flow and contraflow sections. The locations and distances of these locations are detailed in Table 1.

Future Directions

As experiences with contraflow increase and its effectiveness becomes more widely recognized, it is likely that contraflow will be accepted as a standard component emergency preparedness planning and its usage will grow. Several recent high profile negative evacuation experiences have prompted more states to add contraflow options to their response plans. The most notable of these was in Houston, where scores of evacuees (including 23 in a single tragic incident) reportedly perished during the highly criticized evacuation for Hurricane Rita in 2005 [13]. Plans for contraflow are currently under development and should be ready for implementation by the 2007 storm season. Contraflow is also being evaluated for use in some of the larger coastal cities of northeast Australia.

In other locations where hurricanes are not a likely threat, contraflow is also being studied. Some of these examples

include wildfires in the western United States [14] and tsunamis and volcanoes in New Zealand. Greater emphasis on terrorism response have also resulted in cities with few natural hazards to begin examining contraflow for various accidental and purposeful manmade hazards [15].
It is also expected that as contraflow gains in popularity, the application of other developing technologies will be integrated into this strategy. Such has already been the case in South Carolina, Florida, and Louisiana where various intelligent transportation systems (ITS) and other remote sensing technologies have been applied to monitor the state and progression of traffic on contraflow sections during an evacuation. In Washington DC, where reversible flow has been evaluated for use on primary arterial roadways during emergencies, advanced control systems for modifying traffic signal timings have also been studied [16].

Cross References

► Contraflow in Transportation Network
► Dynamic Travel Time Maps

Recommended Reading

1. A Policy on Geometric Design of Highways and Streets, 5th edn., American Association of State Highway and Transportation Officials, Washington, D.C. (2001)
2. Wolshon, B.: Planning and Engineering for the Katrina Evacuation. The Bridge, National Acad. Sci. Eng., vol. 36, no. 1, pp. 27–34 (2006)
3. Wolshon, B.: One-Way-Out: Contraflow Freeway Operation for Hurricane Evacuation. Nat. Hazard. Rev., Am. Soc. Civil Eng. **2**(3), 105–112 (2001)
4. Wolshon, B., Lambert, L.: Convertible Lanes and Roadways, National Cooperative Highway Research Program, Synthesis 340, Transportation Research Board, National Research Council, Washington DC, 92 pp (2004)
5. Wolshon B., Catarella-Michel, A., Lambert, L.: Louisiana Highway Evacuation Plan for Hurricane Katrina: Proactive Management of Regional Evacuations. ASCE J. Transp. Eng. **132**(1), 1–10 (2006)
6. Wolshon B. and McArdle, B.: Temporospatial Analysis of Hurricane Katrina Regional Evacuation Traffic Patterns, ASCE J. Infrastruct. Syst. – Special Infrastructure Planning, Design, and Management for Big Events Issue (in press)
7. Wolshon, B., Urbina, E., Levitan, M., Wilmot, C.: National Review of Hurricane Evacuation Plans and Policies, Part II: Transportation Management and Operations; ASCE Nat. Hazards Rev. **6**(3), 142–161 (2005)
8. United States Army Corps of Engineers: Southeast United States Hurricane Evacuation Traffic Study, performed by Post, Buckeley, Schuh, and Jernigan, Inc., Tallahassee, FL (2000)
9. Theodoulou, G., Contraflow Evacuation on the Westbound I-10 Out of the City of New Orleans. Master's Thesis, Department of Civil and Environmental Engineering, Louisiana State University (2003)
10. Williams, B., Tagliaferri, P.A., Meinhold, S.S., Hummer, J.E., Rouphail, N.M.: Simulation and Analysis of Freeway Lane Reversal for Coastal Hurricane Evacuation, ASCE J. Urban Plann. Dev. – Special Emergency Transportation Issue, **133**(1), 61–72 (2007)
11. Lim, Y.Y.: Modeling and Evaluating Evacuation Contraflow Termination Point Designs. Master's Thesis, Department of Civil and Environmental Engineering, Louisiana State University (2003)
12. Urbina, E., Wolshon, B.: National Review of Hurricane Evacuation Plans and Policies: A Comparison and Contrast of State Practices. Transp. Res., Part A: Policy Pract. **37**(3), 257–275 (2003)
13. Senior Citizens From Houston Die When Bus Catches Fire, Washington Post Staff Writer, Saturday, September 24 (2005); Page A09. [Also available online at: http://www.washingtonpost.com/wp-dyn/content/article/2005/09/23/AR2005092300505.html]
14. Wolshon B. and Marchive, E.: Evacuation Planning in the Urban-Wildland Interface: Moving Residential Subdivision Traffic During Wildfires. ASCE J. Urban Plann. Dev. – Special Emergency Transportation Issue **133**(1), 73–81 (2007)
15. Sorensen, J., Vogt, B.: Interactive Emergency Evacuation Planning Guidebook. Chemical Stockpile Emergency Preparedness Program, Department of Homeland Security. [Available online at: http://emc.ornl.gov/CSEPPweb/evac_files/index.htm] (2006)
16. Chen, M., Chen, L., Miller-Hooks, E.: Traffic Signal Timing for Urban Evacuation, ASCE J. Urban Plann. Dev. – Special Emergency Transportation Issue **133**(1), 30–42

Contraflow in Transportation Network

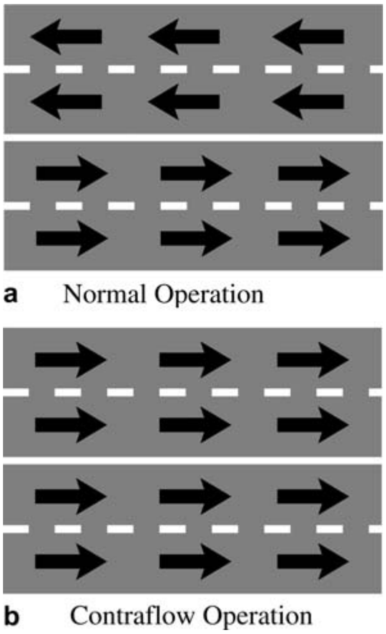
SANGHO KIM
Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA

Synonyms

Lane reversal; Counterflow; Road networks; Networks, spatial; Evacuation routes; Emergency response

Definition

Contraflow is a method designed to increase the capacity of transportation roads toward a certain direction by reversing the opposite direction of road segments. Figure 1 shows a change of road direction under contraflow operation. Contraflow has been primarily used as a part of evacuation schemes. Incoming lanes are reversed to an outbound direction from an affected area during an evacuation to increase the efficiency of traffic movement. When contraflow is implemented on a road network, a significant amount of resources is required in terms of manpower and safety facilities. Automated contraflow execution requires controlled access at both starting and end points of a road. Manual execution requires police officers and barricade trucks. Contraflow planners also need to take into account other factors from the perspectives of planning, design, operation, and financial cost.



Contraflow in Transportation Network, Figure 1 Contraflow road direction

Today, there are many attempts to generate automated contraflow plans with the advanced computing power. However, computerized contraflow involves a combinatorial optimization problem because the number of possible contraflow network configurations is exponentially increasing with the number of road segments (i. e., edges in a graph). In addition, a direction change of a road segment affects the flow (or scheduling) of overall traffic movement at the system level. Thus, it is very hard to find an optimal contraflow network configuration among a huge number of possibilities.

Historical Background

The use of contraflow methods on road networks is not a new concept. Since the beginning of the modern roadway system, there has been a search for solutions to resolve unbalanced flow of traffic due to limited capacity. As the need for massive evacuations of population began to increase around southeastern coastal states threatened by hurricanes, more efficient methods of moving surface transportation have been discussed over the past 20 years [7]. However, the utilization of contraflow has been considered and executed in recent years. During the evacuation of hurricane Floyd in 1999, the South Carolina Department of Transportation measured the traffic flow of Interstate Highway 26 with varying numbers of contraflow lanes [6]. Table 1 summarizes their results. The first

Contraflow in Transportation Network, Table 1 Evacuation traffic flow rates with varying number of contraflow lanes (source: [6], FEMA(2000))

Use configuration	Estimated average outbound flow rate (vehicle/hr)
Normal (two-lanes outbound)	3,000
Normal plus one contraflow lane	3,900
Normal and shoulder plus one contraflow lane	4,200
Normal plus two contraflow lanes	5,000

important finding is that flow rate increases as contraflow lanes (either a lane of opposing traffic or a shoulder) are added. Second, the amount of increased flow per lane is less than the average flow rate under normal condition. It is known that the flow rate per lane under normal operation is about 1,500 vehicles/hr. However, the increased flow rate per lane in the table is under 1,000 vehicles/hr. The limited increases are caused by the unfamiliarity of drivers and their uneasiness driving through an opposite or shoulder lane. Finally, it is observed that the use of shoulder lanes is not as effective as that of normal lanes for contraflow.

During the Rita evacuation in 2005, many evidences showed how ill-planned contraflow negatively affected traffic flow. The following are quoted observations [2] of the traffic problems during the Rita evacuation: "High-occupancy-vehicle lanes went unused, as did many inbound lanes of highways, because authorities inexplicably waited until late Thursday to open some up. ... As congestion worsened state officials announced that contraflow lanes would be established on Interstate Highway 45 (Fig. 2), US Highway 290 and Interstate Highway 10. But by mid-afternoon, with traffic immobile on 290, the plan was dropped, stranding many and prompting others to reverse course. 'We need that route so resources can still get into the city,' explained an agency spokeswoman."

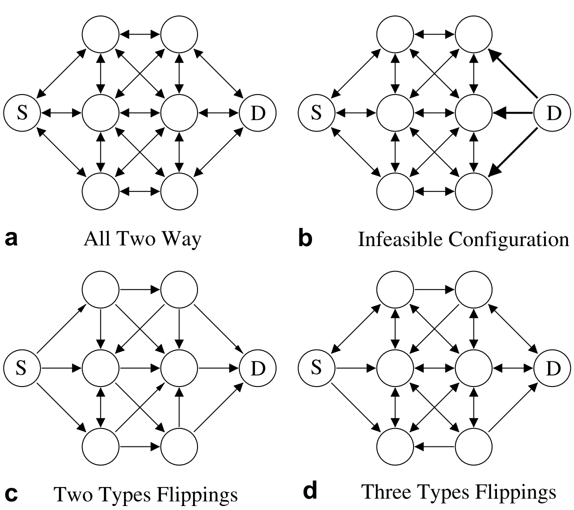
Scientific Fundamentals

Why is Planning Contraflow Difficult?: Figuring out an optimal contraflow network configuration is very challenging due to the combinatorial nature of the problem. Figure 3 shows examples of contraflow network configurations. Suppose that people (e. g., evacuees) in a source node S want to escape to destination node D on the network. Figure 3a is a road network with all edges in two way directions. In other words, no edge is reversed in the network. Figure 3b is an example of a so called 'Infeasible' contraflow configuration because no evacuee can reach destination node D due to the ill-flipped road segments. The network in Fig. 3c allows only two types of

Contraflow in Transportation Network, Figure 2 Hurricane Rita evacuation required contraflow on Interstate Highway 45. Notice that traffic on both sides of I-45 is going north (source: dallasnews.com)

flippings (i. e., ↑, ↓). A network in Fig. 3d allows three types of flippings (i. e., ↑, ↓, ↑↓).

Each network used in these examples has 17 edges. If two types of flippings are allowed as shown in Fig. 3c, the number of possible network configurations is 2^{17}, that is, 131,072. Among them, 89,032 configurations are feasible. An experiment was conducted by assigning some number of evacuees on node S and travel time/capacity attributes on edges. If evacuation time is measured for all feasible configurations (89,023), only 346 configurations have minimum (i. e., optimal) evacuation time, which corresponds to 0.26% out of total possible configurations. For the same network with three types of flippings as shown in Fig. 3d,

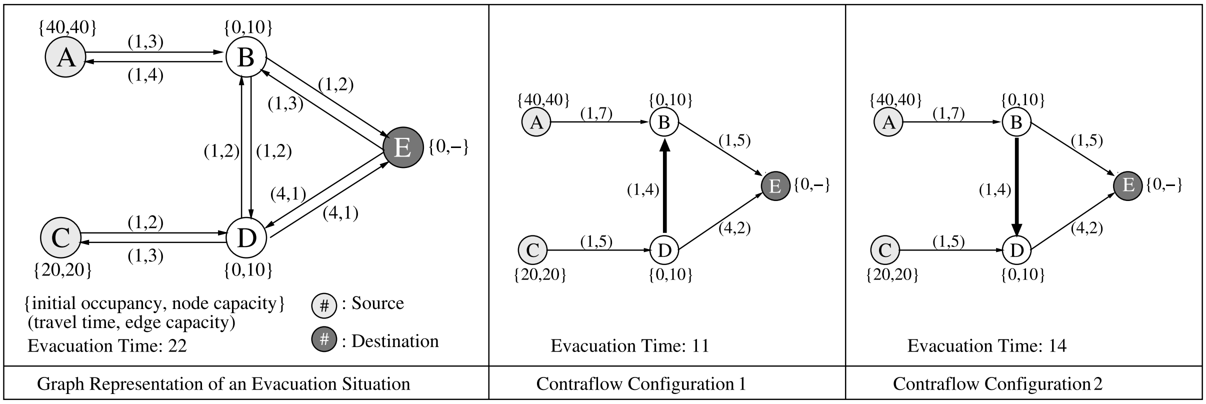


Contraflow in Transportation Network, Figure 3 Examples of infeasible contraflow network and 2 or 3 types of flippings

the number of possible networks is 3^{17}, which is more than 100 million. It is impossible to handle such exponentially large number of configurations even with the most advanced computing system. These examples with such a small size network show why it is difficult to find an optimal contraflow network. The problem is classified as an NP-hard problem in computer science domain.

Modeling Contraflow using Graph: It is often necessary to model a contraflow problem using a mathematical graph. S. Kim et al. [1] presented a modeling approach for the contraflow problem based on graph and flow network. Figure 4 shows a simple evacuation situation on a transportation network. Suppose that each node represents a city with initial occupancy and its capacity, as shown in Fig. 4a. City A has 40 people and also capacity 40. Nodes A and C are modeled as source nodes, while node E is modeled as a destination node (e. g., shelter). Each edge represents a road between two cities with travel time and its capacity. For example, a highway segment between cities A and B has travel time 1 and capacity 3. If a time unit is 5 minutes, it takes 5 minutes for evacuees to travel from A to B and a maximum of 3 evacuees can simultaneously travel through the edge. Nodes B and D have no initial occupancy and only serve as transshipment nodes. The evacuation time of the original network in Fig. 4a is 22, which can be measured using minimum cost flow algorithm.

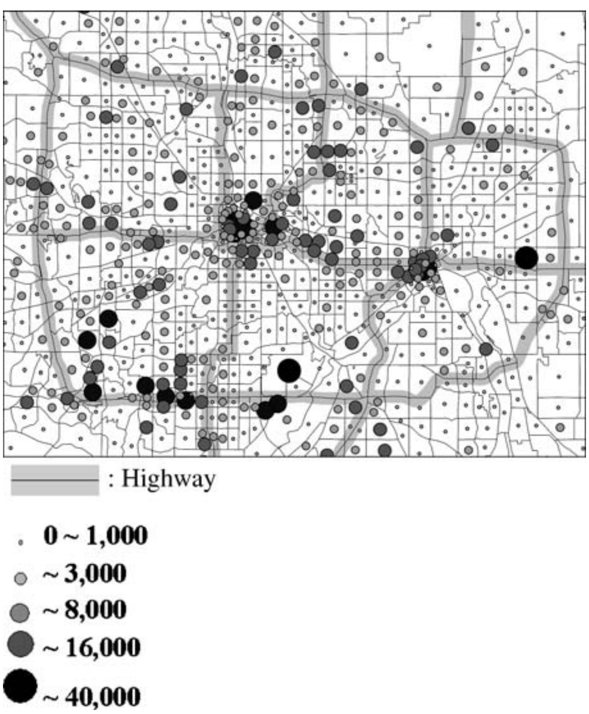
Figure 4b and c illustrate two possible contraflow configurations based on the original graph. All the two-way edges used in the original configuration are merged by capacity and directed in favor of increasing outbound evacuation capacity. There are two candidate configurations that differ in the direction of edges between nodes B and D. If



a b c

Contraflow in Transportation Network, Figure 4 Graph representation of a simple evacuation situation and two following contraflow configuration candidates

the evacuation times of both configurations are measured, the configuration in Fig. 4b has evacuation time 11, while the configuration in Fig. 4c has evacuation time 14. Both configurations not only reduce but also differ in evacuation time. Even though the time difference is just 3 in this example, the difference may be significantly different in the case of a complicated real network. This example illustrates the importance of choice among possible network configurations. In addition, there are critical edges affecting the evacuation time, such as edge (B, D) in Fig. 4.

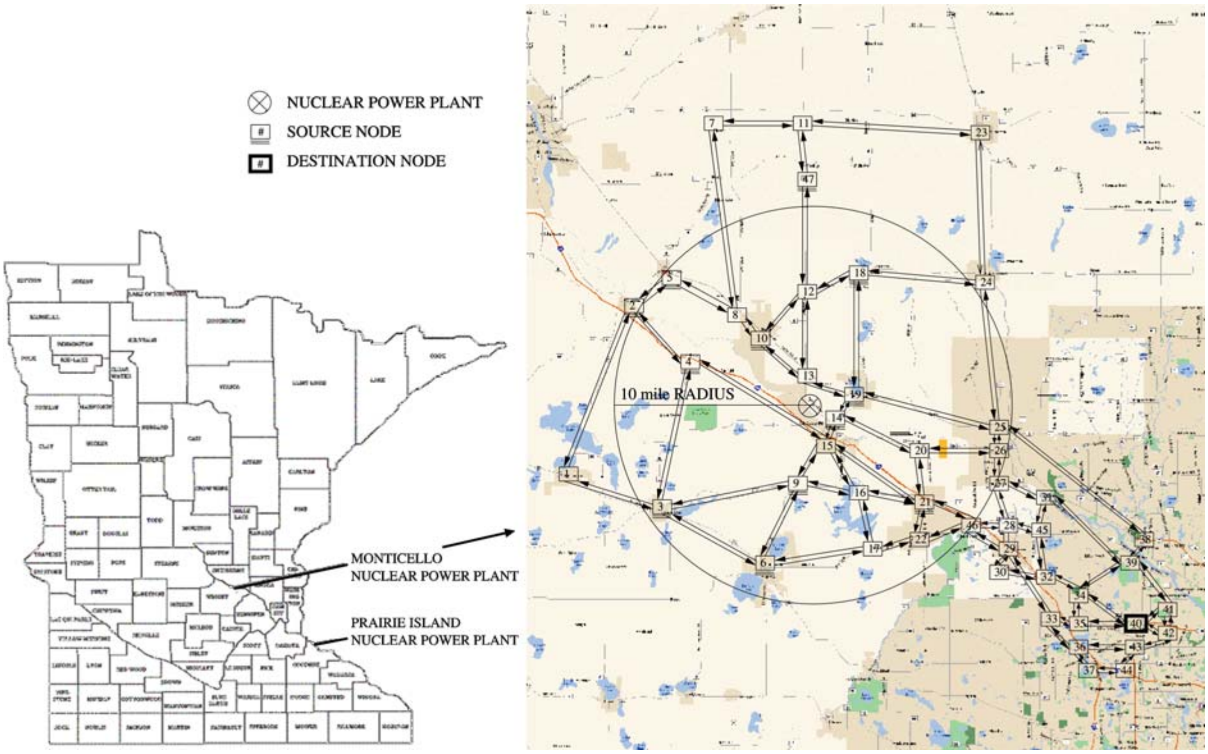


Contraflow in Transportation Network, Figure 5 Day-time population distribution in the Twin Cities, Minnesota

Solutions for Contraflow Planning: S. Kim et al. [1] presented heuristic approaches to find a sub-optimal contraflow network configuration from a given network. Their approaches used the congestion status of a road network to select the most effective target road segments. The experimental results showed that reversing less than 20% of all road segments was enough to reduce evacuation time by more than 40%. Tuydes and Ziliaskopoulos [5] proposed a mesoscopic contraflow network model based on a dynamic traffic assignment method. They formulated capacity reversibility using a mathematical programming method. Theodoulou and Wolshon [4] used CORSIM microscopic traffic simulation to model the freeway contraflow evacuation around New Orleans. With the help of a micro scale traffic simulator, they were able to suggest alternative contraflow configurations at the level of entry and termination points.

Datasets for Contraflow Planning: When emergency managers plan contraflow schemes, the following datasets may be considered. First, population distribution is important to predict congested road segments and to prepare resources accordingly. Figure 5 shows a day-time population distribution in the Twin Cities, Minnesota. The dataset is based on Census 2000 and employment Origin-Destination estimate from the Minnesota Department of Employment and Economic Development, 2002.

Second, a scenario dataset needs to be prepared. The scenario may include road network, accident location, affected area, destination (e. g., evacuation shelter). Figure 6 shows a virtual scenario of a nuclear power plant failure in Monticello, Minnesota. There are twelve cities direct-



Contraflow in Transportation Network, Figure 6 Monticello nuclear power plant located around Twin Cities, Minnesota

ly affected by the failure within 10 miles of the facility and one destination shelter. The affected area is highly dynamic in this case because wind direction can change the shape of the affected area. The road network in the scenario is based on Interstate highway (I-94) and major arterial roads.

Figure 7 shows a possible contraflow scheme based on the Twin Cities population distribution and Monticello nuclear power plant scenario. In this scheme, the dotted road segments represent suggested contraflow. If the suggested road segments are reversed as contraflow, the evacuation time can be reduced by a third from the results of computer simulation.

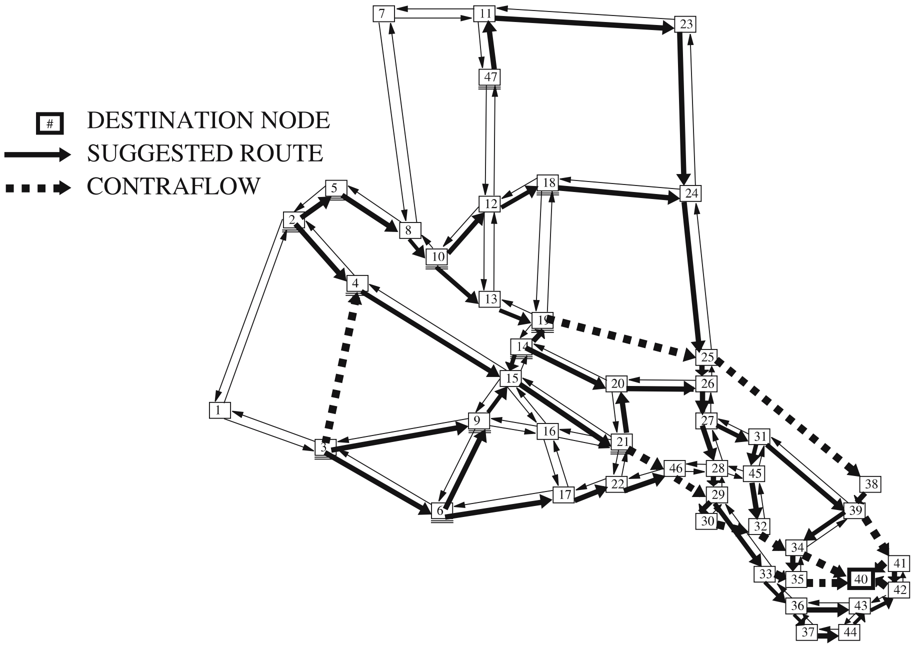
Key Applications

Evacuation under Emergency: When a contraflow program is executed under an emergency situation, several factors should be taken into account: traffic control, accessibility, merging of lanes, use of roadside facilities, safety, labor requirements, and cost [6]. Among these factors, there is a tradeoff between contraflow and safety because most freeways and arterial roads are originally designed for one way direction. An easy example would be a driver who cannot see a traffic light if he drives in the opposite direction. Thus, considerable resources (e. g., police officers, barricade trucks, etc) and cost are required when a contraflow program is implemented. Most coastal states threatened by hurricanes every year prepare contraflow schemes. According to [6], 11 out of 18 coastal states have contraflow plans in place.

The application of contraflow for various desaster types is somewhat limited due to the following reason. Table 2 presents various types of disasters and their properties. According to Litman [2], evacuation route plans should

Type of Disaster	Geographic Scale	Warning	Contraflow Before	Contraflow After
Hurricane	Very large	Days	√	√
Flooding	Large	Days	√	√
Earthquake	Large	None		√
Tsunami	Very large	Short		√
Radiation/toxic release	Small to large	Sometimes		√

Contraflow in Transportation Network, Table 2 Different types of disasters present different types of evacuation properties (source: [2])



Contraflow in Transportation Network, Figure 7 A possible contraflow scheme for Monticello nuclear power plant scenario

take into account the geographic scale and length of warning. Contraflow preparedness is most appropriate for disasters with large geographic scale and long warning time, which gives responders time to dispatch resources and establish reversed lanes. Thus, hurricane and flooding are the most appropriate candidates to apply contraflow plans before disaster. Other types of disasters with relatively short warning time may consider contraflow only after disaster to resolve traffic congestion for back home traffic.

Automated Reversible Lane System: Washington D. C. has an automated reversible lane system to address the daily traffic jams during morning and evening peak time [3]. For example, Interstate Highway 95 operates 28 miles of reversible lanes during 6:00–9:00 AM and 3:30–6:00 PM. The reversible lane system has proven to provide substantial savings in travel time. Reversible lanes are also commonly found in tunnels and on bridges. The Golden Gate Bridge in San Francisco has 2 reversible lanes. Figure 8 shows a controlled access of reversible lane system.

Others: Contraflow programs are used for events with high density population such as football games, concerts, and fireworks on the Fourth of July. Highway construction sometimes requires contraflow. Figure 9 shows an example of contraflow use for highway construction.

Contraflow in Transportation Network, Figure 8 Controlled access of automated reversible lane system in Adelaide (source: wikipedia.org)

Contraflow in Transportation Network, Figure 9 Use of contraflow for highway construction on I-10 in Arizona (source: map.google.com)

Future Directions

For emergency professionals, there are many issues to be solved with regard to contraflow. Currently many planners rely on educated guesses with handcrafted maps to plan contraflow. They need computerized contraflow tools to acquire precise quantification (e. g., evacuation time) of contraflow networks with geographic and demographic data. Other assessments are also required to plan efficient resource use, appropriate termination of contraflow, road markings, post disaster re-entry, etc. For researchers, the development of efficient and scalable contraflow simulation models and tools are urgent tasks. As shown in the Rita evacuation, a large scale evacuation (i. e., three million evacuees) is no longer an unusual event. Efficient tools available in the near future should handle such large scale scenarios with high fidelity traffic models.

Cross References

► Emergency Evacuation, Dynamic Transportation Models
► Emergency Evacuations, Transportation Networks

Recommended Reading

1. Kim, S., Shekhar, S.: Contraflow network reconfiguration for evacuation planning: A summary of results. In: Proceedings of the 13th ACM Symposium on Advances in Geographic Information Systems. pp. 250–259 (2005)
2. Litman, T.: Lessons from katrina and rita: What major disasters can teach transportation planners. J. Trans. Eng. **132**(1), 11–18 (2006)
3. Metropolitan Washington Council of Governments. 2004 performance of regional high-occupancy vehicle facilities on freeways in the washington region. Analysis of Person and Vehicle Volumes. (2005)
4. Theodoulou, G., Wolshon, B.: Alternative methods to increase the effectiveness of freeway contraflow evacuation. J. Trans. Res. Board. Transportation Research Record 1865, 48–56 (2004)
5. Tuydes, H., Ziliaskopoulos, A.: Network re-design to optimize evacuation contraflow. Technical Report 04-4715, Presented at 83rd Annual Meeting of the Transportation Research Board, (2004)
6. Wolshon, B.: One-way-out: Contraflow freeway operation for hurricane evacuation. Nat. Hazard Rev. **2**(3), 105–112 (2001)
7. Wolshon, B., Urbina, E., Levitan, M.: National review of hurricane evacuation plans and policies. Technical report, Hurricane Center, Louisiana State University, Baton Rouge, Louisiana (2002)

Contraint Relations

► Constraint Databases and Data Interpolation

Convergence of GIS and CAD

► Computer Environments for GIS and CAD

Converging

► Movement Patterns in Spatio-temporal Data

Co-Occurrence

► Co-location Pattern Discovery
► Co-location Patterns, Algorithms

Coregistration

► Registration

Co-Registration

► Registration

Correlated

► Patterns, Complex

Correlated Walk

► CrimeStat: A Spatial Statistical Program for the Analysis of Crime Incidents

Correlation Queries

► Correlation Queries in Spatial Time Series Data

Correlation Queries in Spatial Time Series Data

PUSHENG ZHANG
Microsoft Corporation, Redmond, WA, USA

Synonyms

Spatial time series; Correlation queries; Spatial cone tree

Definition

A **spatial framework** consists of a collection of locations and a neighbor relationship. A **time series** is a sequence of observations taken sequentially in time. A **spatial time series dataset** is a collection of time series, each referencing a location in a common spatial framework. For example, the collection of global daily temperature measurements for the last 10 years is a spatial time series dataset over a degree-by-degree latitude-longitude grid spatial framework on the surface of the Earth.

Correlation queries are the queries used for finding collections, e. g. pairs, of highly correlated time series in spatial time series data, which might lead to find potential interactions and patterns. A strongly correlated pair of time series indicates potential movement in one series when the other time series moves.

Historical Background

The massive amounts of data generated by advanced data collecting tools, such as satellites, sensors, mobile devices, and medical instruments, offer an unprecedented opportunity for researchers to discover these potential nuggets of valuable information. However, correlation queries are computationally expensive due to large spatio-temporal frameworks containing many locations and long time sequences. Therefore, the development of efficient query processing techniques is crucial for exploring these datasets.

Previous work on query processing for time series data has focused on dimensionality reduction followed by the use of low dimensional indexing techniques in the transformed space. Unfortunately, the efficiency of these approaches deteriorates substantially when a small set of dimensions cannot represent enough information in the time series data. Many spatial time series datasets fall in this category. For example, finding anomalies is more desirable than finding well-known seasonal patterns in many applications. Therefore, the data used in anomaly detection is usually data whose seasonality has been removed. However, after transformations (e. g., Fourier transformation) are applied to deseasonalize the data, the power spectrum spreads out over almost all dimensions. Furthermore, in most spatial time series datasets, the number of spatial locations is much greater than the length of the time series. This makes it possible to improve the performance of query processing of spatial time series data by exploiting spatial proximity in the design of access methods.

In this chapter, the spatial cone tree, an spatial data structure for spatial time series data, is discussed to illustrate how correlation queries are efficiently supported. The spatial cone tree groups similar time series together based on spatial proximity, and correlation queries are facilitated using spatial cone trees. This approach is orthogonal to dimensionality reduction solutions. The spatial cone tree preserves the full length of time series, and therefore it is insensitive to the distribution of the power spectrum after data transformations.

Scientific Fundamentals

Let $x = \langle x_1, x_2, \ldots, x_m \rangle$ and $y = \langle y_1, y_2, \ldots, y_m \rangle$ be two time series of length m. The correlation coefficient of the two time series is defined as: $\mathrm{corr}(x, y) = \frac{1}{m-1} \sum_{i=1}^{m} (\frac{x_i - \bar{x}}{\sigma_x}) \cdot (\frac{y_i - \bar{y}}{\sigma_y}) = \widehat{x} \cdot \widehat{y}$, where $\bar{x} = \frac{\sum_{i=1}^{m} x_i}{m}$, $\sigma_x = \sqrt{\frac{\sum_{i=1}^{m} (x_i - \bar{x})^2}{m-1}}$, $\bar{y} = \frac{\sum_{i=1}^{m} y_i}{m}$, $\sigma_y = \sqrt{\frac{\sum_{i=1}^{m} (y_i - \bar{y})^2}{m-1}}$, $\widehat{x_i} = \frac{1}{\sqrt{m-1}} \frac{x_i - \bar{x}}{\sigma_x}$, $\widehat{y_i} = \frac{1}{\sqrt{m-1}} \frac{y_i - \bar{y}}{\sigma_y}$, $\widehat{x} = \langle \widehat{x}_1, \widehat{x}_2, \ldots, \widehat{x}_m \rangle$, and $\widehat{y} = \langle \widehat{y}_1, \widehat{y}_2, \ldots, \widehat{y}_m \rangle$.

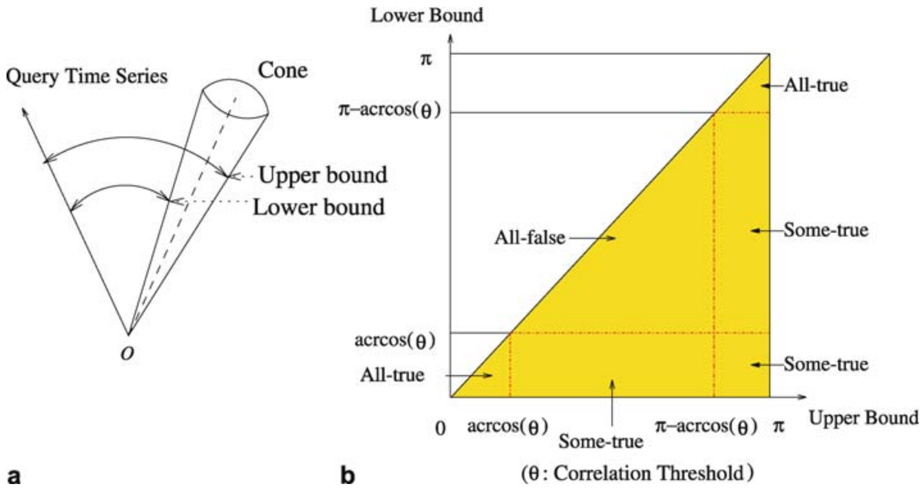
Because the sum of the $\widehat{x_i}^2$ is equal to 1: $\sum_{i=1}^{m} \widehat{x_i}^2 = \sum_{i=1}^{m} (\frac{1}{\sqrt{m-1}} \frac{x_i - \bar{x}}{\sqrt{\frac{\sum_{i=1}^{m} (x_i - \bar{x})^2}{m-1}}})^2 = 1$, $\widehat{x}$ is located in a multi-dimensional unit sphere. Similarly, $\widehat{y}$ is also located in a multi-dimensional unit sphere. Based on the definition of $corr(x, y)$, $corr(x, y) = \widehat{x} \cdot \widehat{y} = \cos(\angle(\widehat{x}, \widehat{y}))$. The correlation of two time series is directly related to the angle between the two time series in the multi-dimensional unit sphere. *Finding pairs of time series with an absolute value of correlation above the user given minimal correlation threshold* θ *is equivalent to finding pairs of time series* $\widehat{x}$ *and* $\widehat{y}$ *on the unit multi-dimensional sphere with an angle in the range of* $[0, \arccos(\theta)]$ *or* $[180° - \arccos(\theta), 180°]$.

A **cone** is a set of time series in a multi-dimensional unit sphere and is characterized by two parameters, the center and the span of the cone. The center of the cone is the mean of all the time series in the cone. The span τ of the cone is the maximal angle between any time series in the cone and the cone center.

A **spatial cone tree** is a spatial data structure for correlation queries on spatial time series data. The spatial cone tree uses a tree data structure, and it is formed of nodes. Each node in the spatial cone tree, except for the root, has one parent node and several-zero or more-child nodes. The root node has no parent. A node that does not have any child node is called a leaf node and a non-leaf node is called an internal node.

A leaf node contains a cone and a data pointer p_d to a disk page containing data entries, and is of the form $\langle$*(cone.span, cone.center)*, $p_d\rangle$. The cone contains one or multiple normalized time series, which are contained in the disk page referred by the pointer p_d. The *cone.span* and *cone.center* are made up of the characteristic parameters for the cone. The data pointer is a block address. An internal node contains a cone and a pointer p_i to an index page containing the pointers to children nodes, and is of the form $\langle$ *(cone.span, cone.center)*, p_i $\rangle$. The *cone.span* and *cone.center* are the characteristic parameters for the cone, which contains all normalized times series in the subtree rooted at this internal node. Multiple nodes are organized in a disk page, and the number of nodes per disk page is defined as the blocking factor for a spatial cone tree. Notice

C



Correlation Queries in Spatial Time Series Data, Figure 1 **a** Upper Bound and Lower Bound, **b** Properties of Spatial Cone Tree

that the blocking factor, the number of nodes per disk page, depends on the sizes of cone span, cone center, and data pointer.

Given a minimal correlation threshold θ ($0 < \theta < 1$), the possible relationships between a cone C and the query time series, T_q, consist of all-true, all-false, or some-true. All-true means that all times series with a correlation over the correlation threshold; all-false means all time series with a correlation less than the correlation threshold; some-true means only part of time series with a correlation over the correlation threshold. The upper bound and lower bound of angles between the query time series and a cone is illustrated in Fig. 1a. Let T is any normalized time series in the cone C and $\angle(\vec{T_q}, \vec{T})$ is denoted for the angle between the query time series vector $\vec{T_q}$ and the time series vector $\vec{T}$ in the multi-dimensional sphere. The following properties are satisfied:

1. If $\gamma_{\max} \in (0, \arccos(\theta))$, then $\angle(\vec{T_q}, \vec{T}) \in (0, \arccos(\theta))$;
2. If $\gamma_{\min} \in (180° - \arccos(\theta), 180°)$, then $\angle(\vec{T_q}, \vec{T}) \in (180° - \arccos(\theta), 180°)$;
3. If $\gamma_{\min} \in (\arccos(\theta), 180°)$ and $\gamma_{\max} \in (\gamma_{\min}, 180° - \arccos(\theta))$, then $\angle(\vec{T_q}, \vec{T}) \in (\arccos(\theta), 180° - \arccos(\theta))$.

If either of the first two conditions is satisfied, the cone C is called an all-true cone (all-true lemma). If the third condition is satisfied, the cone C is called an all-false cone (all-false lemma). If none of the conditions is satisfied, the cone C is called a some-true cone (some-true lemma). These lemma are developed to eliminate cones with all times series satisfying/dissatisfying the correlation threshold in query processing.

The key idea of query processing is to process a correlation query in a filter-and-refine style on the cone level, instead on the individual time series level. The *filtering* step traverses the spatial cone tree, applying the all-true and all-false lemmas on the cones. Therefore, the cones satisfying all-true or all-false conditions are filtered out. The cones satisfying some-true are traversed recursively until all-true or all-false is satisfied or a leaf cone is reached. The *refinement* step exhaustively checks the some-true leaf cones.

Key Applications

The explosive growth of spatial data and widespread use of spatial databases emphasize the need for the automated discovery of spatial knowledge. The complexity of spatial data and intrinsic spatial relationships limits the usefulness of conventional data mining techniques for extracting spatial patterns. Efficient tools for extracting information from geo-spatial data are crucial to organizations which make decisions based on large spatial datasets, including the National Aeronautics and Space Administration (NASA), the National Geospatial-Intelligence Agency (NGA), the National Cancer Institute (NCI), and the United States Department of Transportation (USDOT). These organizations are spread across many application domains including Earth science, ecology and environmental management, public safety, transportation, epidemiology, and climatology. The application of correlation queries used in Earth science is introduced in details as follows.

NASA Earth observation systems currently generate a large sequence of global snapshots of the Earth, including various atmospheric, land, and ocean measurements such as sea surface temperature (SST), pressure, and precipitation. These data are spatial time series data in nature. The climate of the Earth's land surface is strongly influenced by the behavior of the oceans. Simultaneous variations in climate and related processes over widely separated points on the Earth are called teleconnections. For instance, every three to seven years, an El Nino event, i. e., the anomalous warming of the eastern tropical region of the Pacific Ocean, may last for months, having significant economic and atmospheric consequences worldwide. El Nino has been linked to climate phenomena such as droughts in Australia and heavy rainfall along the eastern coast of South America. To investigate such land-sea teleconnections, time series correlation queries across the land and ocean is often used to reveal the relationship of measurements of observations.

Future Directions

In this chapter, the spatial cone tree on spatial time series data was discussed, and how correlation queries can be efficiently supported using the spatial cone tree was illustrated. In future work, more design issues on the spatial cone tree should be further investigated, e. g., the blocking factor and balancing of the tree. The spatial cone tree should be investigated to support complex correlation relationships, such as time lagged correlation. The generalization of spatial cone trees to non-spatial index structures using spherical k-means to construct cone trees is also an interesting research topic.

Recommended Reading

1. National Oceanic and Atmospheric Administration. El Nino Web Page. www.elnino.noaa.gov/
2. Agrawal, R., Faloutsos, C., Swami, A.: Efficient Similarity Search In Sequence Databases. In: Lecture Notes in Computer Science, vol. 730, Berlin/Heidelberg (1993)
3. Box, G., Jenkins, G., Reinsel, G.: Time Series Analysis: Forecasting and Control. Prentice Hall, Upper Saddle River, NJ (1994)
4. Dhillon, I., Fan, J., Guan, Y.: Efficient Clustering of Very Large Document Collections. In: Grossman, R., Kamath, C., Kegelmeyer, P., Kumar, V., Namburu, R. (eds.) Data Mining for Scientific and Engineering Applications. Kluwer Academic Publishers, Dordrecht, The Netherlands (2001)
5. Rafiei, D., Mendelzon, A.: Querying Time Series Data Based on Similarity. IEEE Trans. on Knowl. Data Eng. **12**(5), 675–693 (2000)
6. Chan, F.K., and Fu, A.W.: Haar Wavelets for Efficient Similarity Search of Time-Series: With and Without Time Warping. IEEE Trans. on Knowl. Data Eng. **15**(3), 678–705 (2003)
7. Guttman, A.: R-Trees: A Dynamic Index Structure For Spatial Searching. In: ACM Press, 47–57 (1984)
8. Kahveci, T., Singh, A., Gurel, A.: Similarity Searching for Multi-attribute Sequences. In: IEEE Press, 175 (2002)
9. Keogh, E., Pazzani, M.: An Indexing Scheme for Fast Similarity Search in Large Time Series Databases. In: IEEE Press, 56–67 (1999)
10. Rigaux, P., Scholl, M., Voisard, A.: Spatial Databases: With Application to GIS. Morgan Kaufmann Publishers, Reading (2001)
11. Samet, H.: The Design and Analysis of Spatial Data Structures. Addison-Wesley Publishing Company, San Francisco (1990)
12. Shekhar, S., Chawla, S.: Spatial Databases: A Tour. Prentice Hall, Upper Saddle River (2003), ISBN: 0130174807
13. Zhang, P., Huang, Y., Shekhar, S., Kumar, V.: Correlation Analysis of Spatial Time Series Datasets: A Filter-and-Refine Approach. In: Lecture Notes in Computer Science, vol. 2637, Springer, Berlin/Heidelberg (2003)
14. Zhang, P., Huang, Y., Shekhar, S., Kumar, V.: Exploiting Spatial Autocorrelation to Efficiently Process Correlation-Based Similarity Queries. In: Lecture Notes in Computer Science, vol. 2750, Springer, Berlin/Heidelberg (2003)

COSP

► Error Propagation in Spatial Prediction

Counterflow

► Contraflow in Transportation Network

Coverage Standards and Services, Geographic

WENLI YANG, LIPING DI
Center for Spatial Information Science and Systems,
College of Science, George Mason University,
Fairfax, VA, USA

Definition

A geographic coverage is a representation of a phenomenon or phenomena within a bounded spatiotemporal region by assigning a value or a set of values to each position within the spatiotemporal domain. Geographic coverage standards specify schema and frameworks for geographic coverage or coverage components. Geographic coverage services are those having standard interfaces defined by widely recognized standardization bodies.

Main Text

Geographic phenomena can be observed in two forms, one is discrete and the other is continuous. Discrete phenomena are usually objects that can be directly recognized due to the existence of their geometrical boundaries with other objects. Continuous phenomena usually do not have observable boundaries and vary continuously over space.
The information on the discrete phenomena and that on continuous phenomenon are often used differently and operations performed on the data recording of these two categories of information are usually also different. Thus,

there are often differences in data structure designs, data encoding approaches, data accessing and processing methods for these two types of geographic phenomena. Geographic coverage is a concept for continuous phenomena. Geographic coverage standards defines conceptual schema for coverage and analyze coverage types and components, e. g., [1]. These include characteristics of spatiotemporal domain coverage and attribute range, major coverage types, and operations on coverages. Geographic coverage standards provide a common technology language and guide the development of interoperable services on coverage data. Geographic coverage services perform various functionalities for coverage including collecting, archiving, cataloging, publishing, distributing, and processing of coverage data. Geographic coverage services compliant with standard schema and interfaces are interoperable. They can be described, published and found in standard service catalogues, be accessed by all compliant clients, and be connected in order to construct service chains to accomplish complex geospatial modeling tasks.

Cross References

► Geographic Coverage Standards and Services

References

1. ISO: ISO/TC211, ISO 19123 Geographic information – Schema for coverage geometry and functions (2005)

Crime Mapping

► CrimeStat: A Spatial Statistical Program for the Analysis of Crime Incidents
► Hotspot Detection, Prioritization, and Security

Crime Mapping and Analysis

RONALD E. WILSON, KATIE M. FILBERT
Mapping and Analysis for Public Safety Program & Data Resources, National Institute of Justice, Washington, D.C., MD, USA

Synonyms

Spatial analysis of crime; Spatial aspects of crime; Statistical techniques; Geographical analysis; Environmental criminology; First law of geography; Rational choice; Route activity; Social disorganization

Definition

The term "crime mapping" is inaccurate as it is overly simplistic. Crime mapping is often associated with the simple display and querying of crime data using a Geographic Information System (GIS). Instead, it is a general term that encompasses the technical aspects of visualization and statistical techniques, as well as practical aspects of geographic principles and criminological theories.

From a technical standpoint, the term is a combination of visualization and statistical techniques manifested as software. This combination of techniques is shared between mapping, spatial analysis and spatial data analysis. Mapping is simply a visualization tool that is used to display raw geographic data and output from analysis, which is done through a GIS. Spatial analysis is the statistical testing of geographic features in relation to other geographic features for patterns, or lack there of. Spatial data analysis is the combination of spatial analysis with associated attribute data of the features to uncover spatial interactions between features.

From a practical standpoint, crime mapping is a hybrid of several social sciences, which are geography, sociology and criminology. It combines the basic principles of geographic analysis, sociological and criminological theory and makes it possible to test conjectures from these disciplines in order to confirm or refute them as actual. In essence it has developed into an applied science, with its own tools, that examines a range of issues about society and its relationship with the elements that contribute to crime. Thus, crime mapping is interdisciplinary, involving other disciplines that incorporate the spatial perspectives of social phenomena related to crime, such as inequality, residential stability, unemployment, resource depravation, economic opportunities, housing availability, migration, segregation, and the effects of policy. Using a geographic framework often leads to a more comprehensive understanding of the factors that contribute to or suppress crime.

Even though the term "crime mapping" is a misnomer it will continue to be widely used as a general term with regard to the study of the spatial aspects of crime. Users of the term need to let the context of their work dictate which standpoint is being referred to.

Historical Background

Starting in the 1930s, crime mapping was used with limited success in the United States due to lack of data and the means to analyze that data, computational capacity. Thus, its value was simple depictions on paper of where crimes were occurring. For social science these depictions were not important until researchers from the Chicago School of Sociology combined criminological theory with geographic theory on a map. The result was the theory of social disorganization. Using a map, Shaw and McKay (1942) overlaid residences of juvenile offenders with the Park and Burgess (1925) concentric zone model of urban land uses, including demographic characteristics. They discovered a geographic correlation between impoverished and blighted places with those that had most of the juvenile offender residences. This fostered a new line of research that examined the spatial aspects of crime that spanned from 1950 to the late 1970s. Despite the impact this had on furthering spatial theories of crime there was not much more that could be done because the basic principles of geographic analysis had not yet been operationalized into what geographer Jerome Dobson (1983) would call, "Automated Geography." Personal computers soon came afterwards, but software permitting empirical testing of these theories did not come until much later. It was at this point that crime mapping became useful to law enforcement, primarily to depict where crimes were occurring in order to focus resources (Weisburd and McEwen, 1997). However, there was not yet a relationship between academic institutions and law enforcement agencies to couple theories with actual observations from the street.

Crime mapping with computers made an entrance in the mid 1960s allowing the production of maps of crime by city blocks shaded by volume of incidents. This was still of little interest to researchers studying crime. Even though criminologists were becoming interested in the spatial analysis of crime they were not looking to other disciplines, including geography, for help in analyzing data using a spatial framework. A manifold of software programs from geography were available that could have been used, but there is little evidence in any of the social science literature that demonstrate that these programs were being used. Also neglected were principles of geographic analysis, to analyze the spatial aspects of data. With practitioners, their struggle was different. To produce maps of crime required serious computing infrastructure that, at the time, was only available within larger city government agencies, which did not hold making crime maps in high priority (Weisburd and McEwen, 1997).

The growth of environmental criminology in the 1980s, spearheaded by Paul and Patricia Brantingham, allowed the discipline of geography to make inroads into criminological theory (La Vigne and Groff, 2001). Environmental criminology fused geographical principles and criminological theory together with GIS and provided opportunities to empirically test the theories it was purporting. Significant contributions by George Rengert (1981 and 1989), Jim LeBeau (1987 and 1992) and Keith Harries (1974 and 1980), to environmental criminology using GIS and spatial statistics software continued, thereafter, to strengthen the role of geography in the study of crime. As a result, criminology now has several geographic theories of crime, including rational choice (Cornish and Clarke, 1986), routine activity (Cohen and Felson, 1979), and crime pattern theory (Brantingham and Brantingham, 1981). Social disorganization theory was also extended with geographical principles through the incorporation of simultaneous social interactions between adjacent neighborhoods. For a brief and succinct listing of these theories see Paulsen (2004). At this point, crime mapping branched out to become useful in a new practitioner-based area beyond law enforcement, the criminal justice agency. The confluence of geographic principles, criminological theory and advancing technology led to the development of crime prevention programs based on empirical evidence, such as 'Hot Spot' Policing.

In the late 1980s the Federal government played a role in advancing the use of the crime mapping. The National Institute of Justice (NIJ) funded several efforts under the Drug Market Analysis Program (DMAP) that brought together academic institutions with law enforcement agencies in five cities in the United States (La Vigne and Groff, 2001). The purpose was to identify drug markets and activities associated with them by tracking movement of dealers and users in and out of them. These grants were the first to promote working relationships between practitioners and researchers in the area of crime mapping to move them beyond the limitations each was facing not having the other as a partner.

Continuing improvements in GIS throughout the 1990s, and into the 2000s, made it possible to better assemble, integrate, and create new data. This is probably the greatest impact that GIS has had on crime mapping. Not only could a GIS assemble multiple and disparate sets of demographic, economic and social data with crime data, it could also create new units of analysis that better modeled human behavior. This capability afforded a more accurate understanding of the spatial interactions among offenders, victims and their environments that could be captured and analyzed in ways that more accurately represented human settlement and activity. This freed criminologists from being confined to the standard units of analysis, such as administrative boundaries from the US Census Bureau or other local governmental agencies. GIS provided the unique opportunity to represent boundaries of human activity more accurately through the creation of more distinct partitions, such as police beats or land use, as well as asymmetrical boundaries of human interaction created with buffers or density surfaces. In this regard, there is nothing else like GIS in the study of crime. For practitioners this freed them from having to depend on other government agencies to produce crime maps for law enforcement purposes, as well as provide opportunities to produce custom "on demand" maps for specific purposes, including search warrants or patrol deployment.

The late 1990s saw the advancement of crime mapping in not only both academic departments that study crime and law enforcement agencies, but also in the Federal government. NIJ established the Crime Mapping Research Center in 1997, now the Mapping and Analysis for Public Safety (MAPS) Program, for the purpose of conducting research and evaluation of the spatial aspects of crime. One year later NIJ provided money to the National Law Enforcement and Corrections Technology Center (NLECTC) Rocky Mountain Division to establish the Crime Mapping and Analysis Program (CMAP). This program was to provide assistance to law enforcement and criminal justice agencies specifically in the use of crime mapping. Into the 2000s all large agencies and most medium-sized agencies are using GIS as part of their analysis and operations efforts and are using crime mapping far beyond just the simple mapping of where crime is occurring. Research has continued to refine spatial theories of crime based on better coordination with practitioners, funding from Federal agencies and the development of software for the further understanding of crime through geography.

Scientific Fundamentals

Crime mapping, as an applied science, is ultimately about *where*. As a result, there are contributions from primarily two social science disciplines that make up the foundations of crime mapping. The first provides a set of principles that sets the stage for the study of crime within a spatial framework, geography. The second provides a set of specific spatial theories about criminal activity and environmental conditions that form the foundation of the spatial aspects of crime, criminology.

Geographic Principles

A complete understanding of crime is facilitated by two sets of factors: individual and contextual. Crime mapping deals with the contextual. Therefore, geographic principles are necessary to understand that context. These principles provide a framework for measuring the interactions between places. Analysis in that framework is possible combining long standing geographic principles that have been implemented through GIS and spatial data analysis software. GIS facilitates the visualization of raw data and the results from statistical analysis. Spatial statistical techniques extend traditional statistics to form a more complete approach toward understanding social problems, including crime. The following are the three basic geographic principles that are the foundation for the contextual analysis of crime.

Place

Criminology has a long history of looking at the geographical influences on crime. Some of the most significant pieces of work were in regards to the study of crime in neighborhoods, communities, cites, regions and even across the United States (Brantingham and Brantingham, 1981; Reiss, et al., 1986; Bursik and Grasmick, 1993; Weisburd, et al., 1995). These studies identify "places" in which criminology seeks to understand criminal activity. The focus of studying crime in place demonstrates the use of geography as a framework for contextual analysis that no other discipline can offer. Place becomes the cornerstone because it allows for the categorizing of space by defining a geographic unit of analysis for the systematic measurement of human and environmental characteristics in relation to neighboring places.

Tobler's First Law of Geography

Places are not isolated islands of activity. Interactions, such as social, demographic, or economic occur within and between places. These interactions form spatial relationships based on the concept that those things closer together in space are more related. That is, changes in human activity and physical environments change slowly across space, with abrupt changes being out of the ordinary. Named after Waldo Tobler (1970), this law forms the theoretical foundation for the concept of distance decay that is used for analysis of these spatial interactions and relationships which then allows for measurement in the strength of interactions between places.

Spatial Processes

Human interactions that occur within, and between, geographic places form two concepts: spatial heterogeneity and spatial dependence. Spatial heterogeneity is the variability of human and environmental conditions across space. At the local level this is change across a defined space where conditions, such as racial composition, economic stability, housing conditions, land use, or migration vary. These things are not evenly distributed across space and form various patterns, at different scales, and in multiple directions, all of which are asymmetric. Spatial dependence represents the strength of a relationship of some phenomenon between places that have influence on each other, a concept known as spatial autocorrelation. These patterns range from clusters to randomly distribution to dispersed to uniform. These are indications that human activity and the environments which they develop have a wide range of variability, one that usually follows systemic patterns.

C

Criminological Theories

Criminology has developed a set of spatial theories of crime that have utilized all three of the geographic principles listed.

Rational Choice

Rational choice theory is based on classical ideas that originated in the 1700s, with the work of Cesare Beccaria and others who took a utilitarian view of crime (Beccaria, 1764). This perspective suggests that criminals think rationally and make calculated decisions, weighing costs and risks of committing a crime against potential benefits while being constrained by time, cognitive ability and information available resulting in a 'limited' rather than 'normal' rationality (Cornish and Clarke, 1986). In this sense, rational choice theory also brings in economic ideas and theories into criminology.

Routine Activities

Routine activities theory helps explain why crime occurs at particular places and times. The theory suggests that crime opportunities are a function of three factors that converge in time and place, including a motivated offender, suitable target or victim, and lack of a capable guardian (Cohen and Felson, 1979). A fourth aspect of routine activities theory, suggested by John Eck, is place management. Rental property managers are one example of place managers (Eck and Wartell, 1997). They have the ability to take nuisance abatement and other measures to influence behavior at particular places. Criminals choose or find their targets within context of their routine activities, such as traveling to and from work, or other activities such as shopping, and tend not to go that far out of their way to commit crimes (Felson, 1994).

Crime Pattern

Crime pattern theory looks at the opportunities for crime within context of geographic space, and makes a distinction between crime events and criminality, that is, the propensity to commit crime (Brantingham and Brantingham, 1981). Crime pattern theory integrates rational choice and routine activities theories, with a geographic framework, place. The theory works at various geographic scales, from the macro-level with spatial aggregation at the census tract or other level, to the micro-scale with focus on specific crime events and places. Crime pattern theory focuses on situations or places where there is lack of social control or guardianship over either the suspect or victim, combined with a concentration of targets. For example, a suburban neighborhood can become a hot spot for burglaries because some homes have inadequate protection and nobody home to guard the property.

Social Disorganization

Social disorganization theory emphasizes the importance of social controls in neighborhoods on controlling behavior, particularly for individuals with low self-control or a propensity to commit crime. Social controls can include family, as well as neighborhood institutions such as schools and religious places. When identifying places with social disorganization, the focus is on ability of local residents to control social deviancy (Bursik and Grasmick, 1993). Important factors include poverty, as well as turnover of residents and outmigration, which hinder the development of social networks and neighborhood institutions that lead to collective efficacy (Sampson, Raudenbush, and Earls, 1997).

Key Applications

There are five key applications in crime mapping. These applications are thematic mapping, non-graphical indicators, hot spots, spatial regression and geographic profiling. They make up a full compliment of techniques from elementary to advanced.

Thematic Mapping

Thematic maps are color coded maps that depict the geographic distribution of numeric or descriptive values of some variable. They reveal the geographic patterns of the underlying data. A variable can be quantitative or qualitative. Quantitative maps provide multiple techniques for categorizing the distribution of a variable. Qualitative maps provide a mechanism for classification of some description, or label, of a value. They are often shaded administrative or statistical boundaries, such as census blocks, police beats or neighborhoods. For example, robbery rates based on population can be derived for neighborhood boundaries giving an indication of the neighborhoods that pose the highest risk. However, locations can be symbolized to show quantities based on size or color of the symbol. For example, multiple crime events at a particular location give an indication of repeat victimization, such as common in burglary. However, simple visualization of values and rates can be misleading, especially since the method of classification can change the meaning of a map. Spatial statistics are then used to provide more rigorous and objective analysis of spatial patterns in the data.

Non-Graphical Indicators

Non-graphical statistical tests produce a single number that represents the presence of the clustering of crime incidents or not. These are global level statistics indicating the strength of spatial autocorrelation, but not its location. They compare actual distributions of crime incidents with random distributions. Positive spatial autocorrelation indicates that incidents are clustered, while negative indicates that incidents are uniform. Tests for global spatial autocorrelation within a set of points include Moran's I, (Chakravorty, 1995), Geary's C statistic, and Nearest Neighbor Index (Levine, 2005). After visualizing data in thematic maps these are the first statistical tests conducive to determining whether there are any local level relationships between crime and place exist.

Hot Spots

Hot spots are places with concentrations of high crime or a greater than average level of crime. The converse of a hot spot is a *cold spot*, which are places that are completely, or almost, devoid of crime. Identification and analysis of hot spots is often done by police agencies, to provide guidance as to where to place resources and target crime reduction efforts. Hot spot analysis can work at different geographic levels, from the macro-scale, looking at high crime neighborhoods, or at the micro-scale to find specific places such as particular bars or street segments that are experiencing high levels of crime (Eck et al., 2005). Depending on the level of analysis, police can respond with specific actions such as issuing a warrant or focusing at a neighborhood level to address neighborhood characteristics that make the place more criminogenic. A variety of spatial statistical techniques are used for creating hot spots, such as density surfaces (Levine, 2005), location quotients (Isserman, 1977; Brantingham and Brantingham, 1995; Ratcliffe, 2004), local indicators of spatial autocorrelation (LISA) (Anselin, 1995; Getis and Ord, 1996; Ratcliffe and McCullagh, 1998), and nearest neighborhood hierarchical clustering (Levine, 2005).

Spatial Regression

Regression techniques, such as Ordinary Least Squares (OLS), have been used for quite some time in criminology as explanatory models. This technique has a major limitation, in that it does not account for spatial dependence inherent in almost all data. Holding to geographic principles, a place with high crime is most likely surrounded by neighbors that also experience high crime, thereby displaying spatial autocorrelation, i. e. a spatial effect. Spatial regression techniques, developed by Luc Anselin (2002), take into account spatial dependence in data. Not factoring these spatial effects into models makes them biased and less efficient. Tests have been created for identifying spatial effects in the dependent variable (spatial lag) and among the independent variables (spatial error). If tests detect the presence of spatial lag or error, this form of regression adjusts the model so that spatial effects do not unduly affect the explanatory power of the model.

Geographic Profiling

Geographic profiling is a technique for identifying the likely area where a serial offender resides or other place such as their place of work, that serves as an anchor point. Geographic profiling techniques draw upon crime place theory and routine activities theory, with the assumption that criminals do not go far out of their daily routines to commit crimes. Geographic profiling takes into account a series of crime locations that have been linked to a particular serial criminal and creates a probability surface that identifies the area where the offender's anchor point may be (Rossmo, 2000; Canter, 2003). Geographic profiling was originally developed for use in serial murder, rapes, and other rare but serious crimes. However, geographic profiling is being expanded to high-volume crimes such as serial burglary (Chainey and Ratcliffe, 2005).

Future Directions

The advancement of research and practice in crime mapping rests on continuing efforts in three areas: efforts by research and technology centers, software development, and expansion into law enforcement and criminal justice.

Crime mapping research and technology centers, such as the MAPS Program, the CMAP and the Crime Mapping Center at the Jill Dando Institute (JDI), are primary resources for research, development and application of GIS, spatial data analysis methodologies and geographic technologies. These three centers serves as conduits for much of the work conducted in both the academic and practitioner communities. The MAPS Program is a grant funding and applied research center that serves as a resource in the use of GIS and spatial statistics used in crime studies. The program awards numerous grants for research and development in the technical, applied and theoretical aspects of using GIS and spatial data analysis to study crime, as well as conduct research themselves. As a counterpart to the MAPS Program, CMAP's mission is to serve practitioners in law enforcement and criminal justice agencies by developing tools and training materials for the next generation and crime analysts and applied researchers in the use of GIS and spatial analysis.

In the UK the Jill Dando Institute of Crime Science has a Crime Mapping Center that contributes to the advancement in understanding the spatial aspects of crime with an approach called "crime science." This approach utilizes theories and principles from many scientific disciplines to examine every place as a unique environment for an explanation of the presence or absence of crime. They conduct applied research and provide training with their unique approach on a regular basis. The MAPS Program and the Crime Mapping Center at the JDI hold conferences on a regular basis. These events form the nexus for practitioners and researchers to work together in the exchange of ideas, data, experiences and results from analysis that create a more robust applied science.

Software programs are vital to the progression of the spatial analysis of crime. These programs become the scientific instruments that researchers and practitioners need in understanding human behavior and environmental conditions as they relate to crime. Software, such as CrimeStat, GeoDa and spatial routines for 'R' are being written to include greater visualization capabilities, more sophisticated modeling and mechanisms for seamless operation with other software. For example, in version three of CrimeStat the theory of travel demand was operationalized as a set of routines that apply to criminals as mobile agents in everyday life. GeoDa continues to generate robust tools for visualization based on the principles of Exploratory Data Analysis (EDA). New and cutting edge tools for geographic visualization, spatial statistics and spatial data analysis are being added to the open statistical development environment 'R' on a regular basis. All of these programs provide a rich set of tools for testing theories and discovering new patterns that reciprocally help refine what is known about patterns of crime. The emergence of spatial statistics has proven important enough that even the major statistical software packages, such as SAS, SPSS, and Stata are all incorporating full sets of spatial statistics routines.

The application of crime mapping is expanding into broader areas of law enforcement and criminal justice. In law enforcement mapping is taking agencies in new directions toward crime prevention. For example, the Computer Mapping, Planning and Analysis of Safety Strategies (COMPASS) Program, funded by the NIJ, combines crime data with community data where crime is a characteristic of populations rather than a product. That is to say crime is, at times, a cause of conditions rather than the result of conditions. It is an indicator of the "well being" of neighborhoods, communities or cities. Shared with local level policy makers, COMPASS provides a view into this "well being" of their communities. Resources can be directed to those places that are not "well" and helps to understand what makes other places "well." Combined with problem-oriented policing, a strategy that addresses specific crime problems, this approach can be effective in reducing crime incidents and a general reduction in social disorder (Braga et al., 1999). Coupled with applications in criminal justice, mapping can be utilized to understand the results of policy and the outcomes. This includes topics important to community corrections in monitoring or helping returning offenders, including registered sex offenders. Or, mapping can be of use in allocating probation and parole officers to particular geographic areas, directing probationers and parolees to community services, and selecting sites for new community services and facilities (Karuppannan, 2005). Finally, mapping can even help to understand the geographic patterns of responses to jury summons to determine if there are racial biases are occurring in some systematic way across a jurisdiction (Ratcliffe, 2004).

These three elements will persist and intertwine to evermore incorporate the geographic aspects of basic and applied research of crime through technology. The advancement of knowledge that crime mapping can provide will require continued reciprocation of results between research and practice through technology (Stokes, 1997). The hope is that researchers will continue to create new techniques and methods that fuse classical and spatial statistics together to further operationalize geographic principles and criminological theory to aid in the understanding of crime. Practitioners will implement new tools that are developed for analyzing crime with geographic perspectives. They will also continue to take these tools in new directions as "improvers of technology" (Stokes, 1997) and discover new patterns as those tools become more complete in modeling places.

Cross References

- ► Autocorrelation, Spatial
- ► CrimeStat: A Spatial Statistical Program for the Analysis of Crime Incidents
- ► Data Analysis, Spatial
- ► Exploratory Visualization
- ► Hotspot Detection, Prioritization, and Security
- ► Patterns, Complex
- ► Spatial Econometric Models, Prediction
- ► Spatial Regression Models
- ► Statistical Descriptions of Spatial Patterns
- ► Time Geography
- ► Visualizing Constraint Data

Acknowledgements

We would like to thank Keith Harries, Dan Helms, Chris Maxwell and Susan Wernicke-Smith for providing comments on this entry in a very short time. They provid-

ed valuable comments that were used toward crafting this entry.
The views expressed in this paper are those of the authors, and do not represent the official positions or policies of the National Institute of Justice or the US Department of Justice.

Recommended Reading

Anselin, L.: Local indicators of spatial association – LISA. Geogr. Anal. **27**, 93–115 (1995)

Anselin, L.: Under the Hood: Issues in the Specification and Interpretation of Spatial Regression Models. http://sal.uiuc.edu/users/anselin/papers.html (2002)

Braga, A.A., Weisburd, D.L., et al.: Problem-oriented policing in violent crime places: A randomized controlled experiment. Criminology **7**, 541–580 (1999)

Brantingham, P., Brantingham, P.: Environmental Criminology. Waverland Press, Prospect Heights, Il (1981)

Brantingham, P., Brantingham, P.: Location quotients and crime hotspots in the city. In: Block, C., Dabdoub, M., Fregly, S. (eds.) Crime analysis through computer mapping. Police Executive Research Forum, Washington DC (1995)

Beccaria, C.: in Richard Davies, translator: On Crimes and Punishments, and Other Writings. Cambridge University Press (1764)

Bursik, R.J., Grasmick, H.G.: Neighborhoods and Crime: The Dimensions of Effective Community Control. Lexington Books, New York, NY (1993)

Canter, D.: Mapping Murder: The Secrets of Geographic Profiling. Virgin Publishing, London (2003)

Chainey, S., Ratcliffe, J.: GIS and Crime Mapping. John Wiley and Sons, Ltd., Hoboken, NJ (2005)

Chakravorty, S.: Identifying crime clusters: The spatial principles, Middle States Geogr. **28**, 53–58 (1995)

Clarke, R.V.: Situational Crime Prevention: Successful Case Studies. Harrow and Heston, New York (1992)

Cohen, L., Felson, M.: Social Change and Crime Rate Trends. Am. Soc. Rev. **44** (1979)

Cornish, D., Clarke, R.V.: The Reasoning Criminal. Springer-Verlag (1986)

Cresswell, T.: Place: A Short Introduction. Blackwell Publishing Ltd (2004)

Dobson, J. E.: Automated geography. The Professional Geogr. **35**(2), 135–143 (1983)

Eck, J., Chainey, S., Cameron, J., Leitner, M., Wilson, R.E.: Mapping Crime: Understanding Hot Spots. Washington, DC; National Institute of Justice (2005)

Eck, J., Wartell, J.: Reducing Crime and Drug Dealing by Improving Place Management: A Randomized Experiment. National Institute of Justice (1997)

Felson, M.: Crime and Everyday Life. Pine Forge (1994)

Getis, A., Ord, J.K.: Local spatial statistics: an overview. In: Longley, P., Batty, M. (eds.): Spatial Analysis: Modelling in a GIS Environment, pp. 261–277. Geoinformation International, Cambridge, England (1996)

Haining, R.: Spatial Data Analysis: Theory and Practice. Cambridge University Press (2003)

Harries, K.D.: The Geography of Crime and Justice. McGraw-Hill, New York, NY (1974)

Harries, K.D.: Crime and the Environment. Charles C Thomas Press, Spingfield, Il (1980)

Isserman, A.M.: The Location Quotient Approach for Estimating Regional Economic Impacts. J. Am. Inst. Plann. **43**, 33–41 (1977)

Karuppannan, J.: Mapping and Corrections: Management of Offenders with Geographic Information Systems. Corrections Compendium. http://www.iaca.net/Articles/drjaishankarmaparticle.pdf (2005)

La Vigne, N. G., Groff, E. R.: The Evolution of Crime Mapping in the United States: From the descriptive to the analytic. In: Hirschfield, A., Bowers, K. (eds.) Mapping and Analyzing Crime Data (pp. 203–221). University of Liverpool Press, Liverpool, UK (2001)

LeBeau, J.L.: Patterns of Stranger and Serial Rape Offending: Factors Distinguishing Apprehended and At Large Offenders. J. Crim. Law Criminol. **78**(2), 309–326 (1987)

LeBeau, J.L.: Four Case Studies Illustrating the Spatial-temporal Analysis of Serial Rapists. Police Stud. **15**, 124–145 (1992)

Levine, N.: CrimeStat III Version 3.0, A Spatial Statistics Program for the Analysis of Crime Incident Locations (2005)

Park, R.E., Burgess, E.W., McKenzie, R.D.: The City: Suggestions for Investigation of Human Behavior in the Urban Environment. University of Chicago Press, Chicago, Il (1925)

Paulsen, D. J., Robinson, M.B.: Spatial Aspects of Crime: Theory and Practice. Allyn and Bacon, Boston, MA (2004)

Ratcliffe, J.H., McCullagh, M.J.: The Perception of Crime Hotspots: A spatial study in Nottingham, UK. In: Crime Mapping Case Studies: Successes in the Field. National Institute of Justice, Washington, DC (1998)

Ratcliffe, J.H.: Location Quotients and Force-field Analysis. 7th Annual International Crime Mapping Research Conference (2004)

Ray, J.C.: Crime Prevention through Environmental Design. Sage Publications, Beverly Hills, CA (1977)

Reiss, A.J., Tonry, M. (eds.): Communities and Crime, Vol 8. University of Chicago Press, Chicago, Il (1986)

Rengert, G.F., Simon, H.: Crime Spillover. Ca; Sage Publications, Beverley Hills (1981)

Rengert, G.F.: Behavioral Geography and Criminal Behavior. In: Evans, D.J., Herbert, D.T. (eds.) The Geography of Crime. Routledge, London (1989)

Ronald, C.V.: Situational Crime Prevention. Harrow and Heston (1992)

Rossmo, D.K.: Geographic Profiling. Boca Raton, FL: CRC Press (2000)

Sampson, R.J., Raudenbush, S.R., Earls, F.: Neighborhoods and Violent Crime: A Multilevel Study of Collective Efficacy. Science **227**, 918–924 (1997)

Shaw, C.R., McKay, H.D.: Juvenile Delinquency and Urban Areas. University of Chicago Press, Chicago, Il (1942)

Stokes, D.E.: Pasteur's Quadrant: Basic Science and Technological Innovation. Brookings Institution Press, Washington, DC (1997)

Tobler, W.R.: A Computer Movie Simulating Urban Growth in the Detroit Region. Econ. Geogr. **46**, 234–240 (1970)

Weisburd, D., Eck, J.E. (eds.): Crime and Place: Crime Prevention Studies, Vol. 4. Police Executive Research Forum: Willow Tree Press (1995)

Weisburd, D., Green, L.: Policing Drug Hot Spots: The Jersey City Drug Market Analysis Experiment. Justice Q. **12**(4), 711–736 (1995)

Weisburd, D., McEwen, T. (eds.): Introduction: Crime Mapping and Crime Prevention. In: Crime Mapping and Crime Prevention (pp. 1–23). Criminal Justice Press, Monsey, NY (1997)

Crime Travel Demand

► CrimeStat: A Spatial Statistical Program for the Analysis of Crime Incidents

CrimeStat

► CrimeStat: A Spatial Statistical Program for the Analysis of Crime Incidents

CrimeStat: A Spatial Statistical Program for the Analysis of Crime Incidents

NED LEVINE
Ned Levine & Associates, Houston, TX, USA

Synonyms

CrimeStat; Spatial statistics program; Crime mapping; Hotspot; Centrographic measures; Interpolation; Space-time interaction; Knox test; Mantel test; Correlated walk; Journey to crime analysis; Geographic profiling; Crime travel demand

Definition

CrimeStat is a spatial statistics and visualization program that interfaces with desktop GIS packages. It is a stand-alone Windows program for the analysis of crime incident locations and can interface with most desktop GIS programs. Its aim is to provide statistical tools to help law enforcement agencies and criminal justice researchers in their crime mapping efforts. The program has many statistical tools, including centrographic, distance analysis, hot spot analysis, space-time analysis, interpolation, Journey-to-Crime estimation, and crime travel demand modeling routines. The program writes calculated objects to GIS files that can be imported into a GIS program, including shape, MIF/MID, BNA, and ASCII. The National Institute of Justice is the distributor of CrimeStat and makes it available for free to analysts, researchers, educators, and students.[1] The program is distributed along with a manual that describes each of the statistics and gives examples of their use [1].

Historical Background

CrimeStat has been developed by Ned Levine and Associates since the late 1990s under grants from the National Institute of Justice. It is an outgrowth of the Hawaii Pointstat program that was UNIX-based [2]. CrimeStat, on the other hand, is a Windows-based program. It is written in C++ and is multi-threading. To date, there have been three major versions with two updates. The first was in 1999 (version 1.0) with an update in 2000 (version 1.1). The second was in 2002 (CrimeStat II) and the third was in 2004 (CrimeStat III). The current version is 3.1 and was released in March 2007.

[1]The program is available at http://www.icpsr.umich.edu/crimestat.

Scientific Fundamentals

The current version of CrimeStat covers seven main areas of spatial analysis: centrographic; spatial autocorrelation, hot spot analysis, interpolation, space-time analysis, Journey-to-Crime modeling, and crime travel demand modeling.

Centrographic Measures

There are a number of statistics for describing the general properties of a distribution. These include central tendency of the overall spatial pattern, dispersion and directionality. Among the statistics are the mean center, the center of minimum distance, the standard distance deviation, the standard deviational ellipse, the harmonic mean, the geometric mean, and the directional mean [3].

Spatial Autocorrelation

There are several statistics for describing spatial autocorrelation, including Moran's I, Geary's C, and a Moran Correlogram [4,5,3]. There are also several statistics that describe spatial autocorrelation through the properties of distances between incidents including the nearest neighbor statistic [6], the linear nearest neighbor statistic, the K-order nearest neighbor distribution [7], and Ripley's K statistic [8]. The testing of significance for Ripley's K is done through a Monte Carlo simulation that estimates approximate confidence intervals.

Hot Spot Analysis

An extreme form of spatial autocorrelation is a *hot spot*. While there is no absolute definition of a 'hot spot', police are aware that many crime incidents tend to be concentrated in a limited number of locations. The Mapping and Analysis for Public Safety Program at the National Institute of Justice has sponsored several major studies on crime hot spot analysis [9,10,11].
CrimeStat includes seven distinct 'hot spot' analysis routines: the mode, the fuzzy mode, nearest neighbor hierarchical clustering [12], risk-adjusted nearest neighbor hierarchical clustering [13], the Spatial and Temporal Analy-

sis of Crime routine (STAC) [14], K-means clustering, and Anselin's Moran statistic [15].

The *mode* counts the number of incidents at each location. The *fuzzy mode* counts the number of incidents at each location within a specified search circle; it is useful for detecting concentrations of incidents within a short distance of each other (e. g., at multiple parking lots around a stadium; at the shared parking lot of multiple apartment buildings).

The *nearest neighbor hierarchical clustering* routine defines a search circle that is tied to the random nearest neighbor distance. First, the algorithm groups incidents that are closer than the search circle and then searches for a concentration of multiple incidents within those selected. The center of each concentration is identified and all incidents within the search circle of the center of each concentration are assigned to the cluster. Thus, incidents can belong to one-and-only-one cluster, but not all incidents belong to a cluster. The process is repeated until the distribution is stable (first-order clusters). The user can specify a minimum size for the cluster to eliminate very small clusters (e. g., 2 or 3 incidents at the same location). Once clustered, the routine then clusters the first-order clusters to produce second-order clusters. The process is continued until the grouping algorithm fails. The *risk-adjusted nearest neighbor hierarchical clustering* routine follows the same logic but compares the distribution of incidents to a baseline variable. The clustering is done with respect to a baseline variable by calculating a cell-specific grouping distance that would be expected on the basis of the baseline variable, rather than a single grouping distance for all parts of the study area.

The *Spatial and Temporal Analysis of Crime* hot spot routine (STAC) is linked to a grid and groups on the basis of a minimum size. It is useful for identifying medium-sized clusters. The *K-means* clustering algorithm divides the points into K distinct groupings where K is defined by the user. Since the routine will frequently create clusters of vastly unequal size due to the concentration of incidents in the central part of most metropolitan areas, the user can adjust them through a separation factor. Also, the user can define specific starting points (seeds) for the clusters as opposed to allowing the routine to find its own.

Statistical significance of these latter routines is tested with a Monte Carlo simulation. The nearest neighbor hierarchical clustering, the risk-adjusted nearest neighbor hierarchical clustering, and the STAC routines each have a Monte Carlo simulation that allows the estimation of approximate confidence intervals or test thresholds for these statistics.

Finally, unlike the other hot spot routines, *Anselin's Local Moran* statistic is applied to aggregates of incidents in zones. It calculates the similarity and dissimilarity of zones relative to nearby zones by applying the Moran's I statistic to each zone. An approximate significance test can be calculated using an estimated variance.

Interpolation

Interpolation involves extrapolating a density estimate from individual data points. A fine-mesh grid is placed over the study area. For each grid cell, the distance from the center of the cell to each data point is calculated and is converted into a density using a mathematical function (a kernel). The densities are summed over all incidents to produce an estimate for the cell. This process is then repeated for each grid cell [16]. *CrimeStat* allows five different mathematical functions to be used to estimate the density. The particular dispersion of the function is controlled through a bandwidth parameter and the user can select a fixed or an adaptive bandwidth. It is a type of hot spot analysis in that it can illustrate where there are concentrations of incidents. However it lacks the precision of the hot spot routines since it is smoothed. The hot spot routines will show exactly which points are included in a cluster.

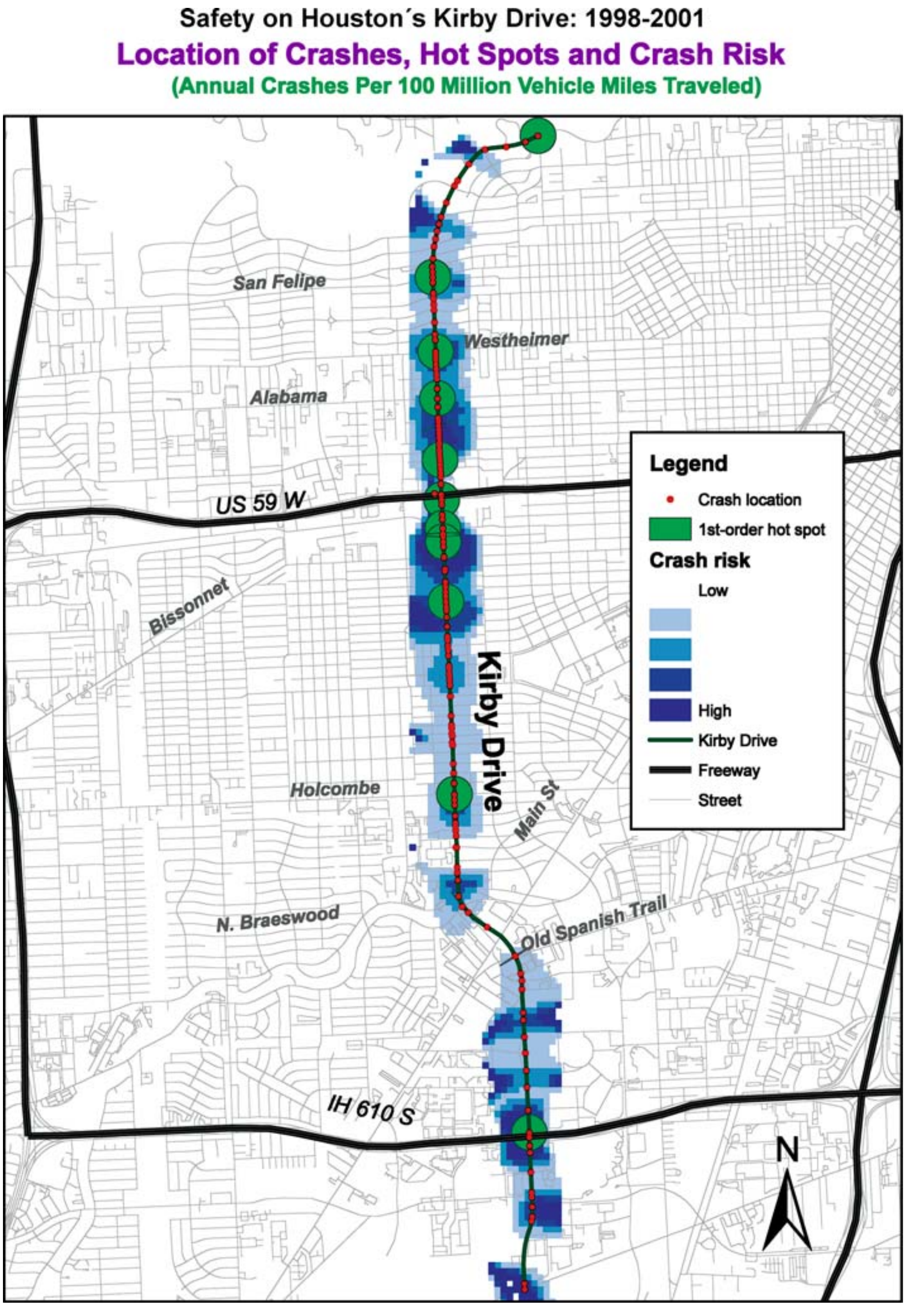
CrimeStat has two different kernel function, a single-variable kernel density estimation routine for producing a surface or contour estimate of the density of incidents (e. g., the density of burglaries) and a dual-variable kernel density estimation routine for comparing the density of incidents to the density of an underlying baseline (e. g., the density of burglaries relative to the density of households).

As an example, Fig. 1 shows motor vehicle crash risk along Kirby Drive in Houston for 1999–2001. *Crash risk* is defined as the annual number of motor vehicle crashes per 100 million vehicle miles traveled (VMT) and is a standard measure of motor vehicle safety. The duel-variable kernel density routine was used to estimate the densities with the number of crashes being the incident variable and VMT being the baseline variable. In the map, higher crash risk is shown as darker. As a comparison, hot spots with 15 or more incidents were identified with the nearest neighbor hierarchical clustering routine and are overlaid on the map as are the crash locations.

Space-Time Analysis

There are several routines for analyzing clustering in time and in space. Two are global measures – the *Knox* and *Mantel* indices, which specify whether there is a relationship between time and space. Each has a Monte Carlo simulation to estimate confidence intervals around the calculated statistic.

The third space-time routine is a specific tool for predicting the behavior of a serial offender called the *Correlated*



CrimeStat: A Spatial Statistical Program for the Analysis of Crime Incidents, Figure 1 Safety on Houston's Kirby Drive: 1998–2001

Walk Analysis module. This module analyzes periodicity in the sequence of events committed by the serial offender by distance, direction, and time interval. It does this by analyzing the sequence of lagged incidents. A diagnostic correlogram allows the user to analyze periodicity by different lags. The user can then specify one of several methods for predicting the next incident that the serial offender will commit, by location and by time interval. Error is, of course, quite sizeable with this methodology because serial offenders don't follow strict mathematical rules. But the method can be useful for police because it can indicate whether there are any repeating patterns that the offender is following.

Journey-to-Crime Analysis

A useful tool for police departments seeking to apprehend a serial offender is *Journey-to-crime analysis* (sometimes known as *Geographic Profiling*). This is a method for estimating the likely residence location of a serial offender given the distribution of incidents and a model for travel distance [17,18,19,20]. The method depends on building a typical travel distance function, either based on empirical distances traveled by known offenders or on an *a priori* mathematical function that approximates travel behavior (e. g., a negative exponential function, a negative exponential function with a low use 'buffer zone' around the offender's residence).

CrimeStat has a Journey-to-Crime routine that uses the travel distance function and a Bayesian Journey-to-Crime routine that utilizes additional information about the likely origins of offenders who committed crimes in the same locations. With both types – the traditional distance-based and the Bayesian, there are both calibration and estimation routines. In the calibration routine for the Journey-to-Crime routine, the user can create an empirical travel distance function based on the records of known offenders where both the crime location and the residence location were known (typically from arrest records). This function can then be applied in estimating the likely location of a single serial offender for whom his or her residence location is not known.

The Bayesian Journey-to-Crime routine utilizes information about the origins of other offenders who committed crimes in the same locations as a single serial offender. Again, based on a large set of records of known offenders, the routine estimates the distribution of origins of these offenders. This information can then be combined with the travel distance function to make estimates of the likely location of a serial offender where the residence location is not known. Early tests of this method suggest that it is 10–15% more accurate than the traditional travel distance only method in terms of estimating the distance between the highest probability location and the location where the offender lived.

As an example, Fig. 2 shows a Bayesian probability model of the likely residence location of a serial offender who committed five incidents between 1993 and 1997 in Baltimore County, Maryland (two burglaries and three larceny thefts). The grid cell with the highest probability is outlined. The location of the incidents is indicated as is the actual residence location of the offender when arrested. As seen, the predicted highest probability location is very close to the actual location (0.14 of a mile error).

Crime Travel Demand Modeling

CrimeStat has several routines that examine travel patterns by offenders. There is a module for modeling crime travel behavior over a metropolitan area called *Crime Travel Demand* modeling. It is an application of travel demand modeling that is widely used in transportation planning [21]. There are four separate stages to the model. First, predictive models of crimes occurring in a series of zones (crime destinations) and originating in a series of zones (crime origins) are estimated using a non-linear (Poisson) regression model with a correction for over-dispersion [22]. Second, the predicted origins and destinations are linked to yield a model of crime *trips* from each origin zone to each destination zone using a gravity-type spatial interaction model. To estimate the coefficients, the calibrated model is compared with an actual distribution of crime trips.

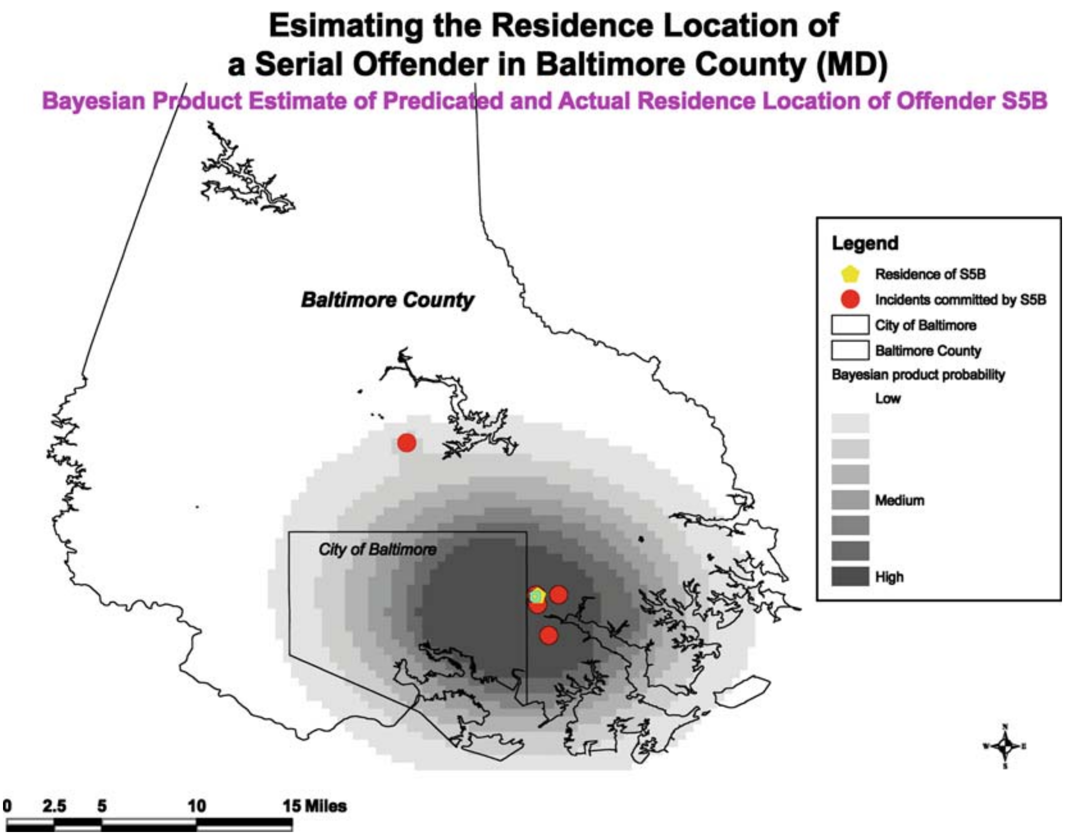
In the third stage, the predicted crime trips are separated into different travel modes using an approximate multinomial utility function [23]. The aim is to examine possible strategies used by offenders in targeting their victims. Finally, the predicted crime trips by travel mode are assigned to particular routes, either on a street network or a transit network. The cost of travel along the network can be estimated using distance, travel time, or a generalized cost using the A* shortest path algorithm [24].

Once calibrated, the model can be used to examine possible interventions or policy scenarios. For example, one study examined the travel behavior of individuals who were involved in Driving-while-Intoxicated (DWI) motor vehicle crashes in Baltimore County. Neighborhoods where a higher proportion of DWI drivers involved in crashes were identified as were locations where many DWI crashes had occurred. Interventions in both high DWI driver neighborhoods and the high DWI crash locations were simulated using the model to estimate the likely reduction in DWI crashes that would be expected to occur if the interventions were actually implemented.

Key Applications

CrimeStat is oriented mostly toward the law enforcement and criminal justice fields, but it has been used widely by researchers in other fields including geography, traffic safety, urban planning, sociology, and even fields like botany and forestry. The tools reflect a range of applications that criminal justice researchers and crime analysts might find useful, some describing the spatial distribution and others being targeted to particular offenders.

For example, hot spot analysis is particularly useful for police departments. Police officers, crime analysts and researchers are very familiar with the concentration of crime or other incidents that occur in small areas. Further they are aware that many offenders live in certain neighborhoods that are particularly poor and lacking in social amenities. There is a large literature on high crime



CrimeStat: A Spatial Statistical Program for the Analysis of Crime Incidents, Figure 2 Estimating the residence location of a serial offender in Baltimore County (MD)

areas so that the phenomenon is very well known (e. g., see [25,26]). The hot spot tools can be useful to help police systematically identify the high crime areas as well as the areas where there are concentrations of offenders (which are not necessarily the same as the high crime locations). For example, the hot spot tools were used to identify locations with many red light running crashes in Houston as a prelude for introducing photo-enforcement. The Massachusetts State Police used the neighbor nearest hierarchical clustering algorithm to compare heroin and marijuana arrest locations with drug seizures in one small city [27].

Another criminal justice application is the desire to catch serial offenders, particularly high visibility ones. The Journey-to-Crime and Bayesian Journey-to-Crime routines can be useful for police departments in that it can narrow the search that police have to make to identify likely suspects. Police will routinely search through their database of known offenders; the spatial narrowing can reduce that search substantially. The CrimeStat manual has several examples of the Journey-to-Crime tool being used to identify a serial offender. As an example, the Glendale (Arizona) Police Department used the Journey-to-Crime routine to catch a felon who had committed many auto thefts [28].

Many of the other tools are more relevant for applied researchers such as the tools for describing the overall spatial distribution or for calculating risk in incidents (police typically are interested in the volume of incidents) or for modeling the travel behavior of offenders. Two examples from the CrimeStat manual are given. First, the spatial distribution of "*Man With A Gun*" calls for service during Hurricane Hugo in Charlotte, North Carolina was compared with a typical weekend [29]. Second, the single-variable kernel density routine was used to model urbanization changes in the Amazon between 1996 and 2000 [30].

Future Directions

Version 4 of CrimeStat is currently being developed (CrimeStat IV). The new version will have a complete restructuring to modernize it consistent with trends in computer science. First, there will be a new GUI interface that will be more Windows Vista-oriented. Second, the code is being revised to be consistent with the .NET framework and selected routines will be compiled as objects in a library that will be available for programmers and third-party applications. Third, additional statistics relevant for crime prediction are being developed. These include a spatial regression module using Markov Chain Monte Carlo methods and an incident detection module for identifying emerging crime hot spot spots early in their sequence. Version 4 is expected to be released early in 2009.

Cross References

- ▶ Autocorrelation, Spatial
- ▶ Crime Mapping and Analysis
- ▶ Data Analysis, Spatial
- ▶ Emergency Evacuation, Dynamic Transportation Models
- ▶ Hotspot Detection, Prioritization, and Security
- ▶ Movement Patterns in Spatio-temporal Data
- ▶ Nearest Neighbors Problem
- ▶ Patterns in Spatio-temporal Data
- ▶ Public Health and Spatial Modeling
- ▶ Routing Vehicles, Algorithms
- ▶ Sequential Patterns, Spatio-temporal
- ▶ Statistical Descriptions of Spatial Patterns

Recommended Reading

1. Levine, N.: CrimeStat III: A Spatial Statistics Program for the Analysis of Crime Incident Locations (version 3.1). Ned Levine & Associates, Houston, TX and the National Institute of Justice, Washington, DC. March (2007)
2. Levine, N.: Spatial statistics and GIS: software tools to quantify spatial patterns. Journal of the American Planning Association **62**(3), 381–392 (1996)
3. Ebdon, D.: Statistics in Geography (2nd edn. with corrections). Blackwell, Oxford (1988)
4. Moran, P.A.P.: The interpretation of statistical maps. J. Roy. Stat. Soc. B **10**, 243–251 (1948)
5. Geary, R.: The contiguity ratio and statistical mapping. The Incorporated Statistician **5**, 115–145 (1954)
6. Clark, P.J., Evans, F.C.: Distance to nearest neighbor as a measure of spatial relationships in populations. Ecology **35**, 445–453 (1954)
7. Cressie, N.: Statistics for Spatial Data. John Wiley & Sons, Inc, New York (1991)
8. Ripley, B.D.: Spatial Statistics. John Wiley & Sons, New York (1981)
9. Harries, K.: Mapping Crime: Principle and Practice. NCJ 178919, National Institute of Justice, US Department of Justice: Washington, DC. Available at http://www.ncjrs.org/html/nij/mapping/pdf.html (1999)
10. LaVigne, N., Wartell, J.: Crime Mapping Case Studies: Success in the Field (vol. 1). Police Executive Research Forum and National Institute of Justice, US Department of Justice: Washington, DC (1998)
11. Eck, J., Chainey, S., Cameron, J., Leitner, M., Wilson, R.E.: Mapping Crime: Understanding Hot Spots. Mapping and Analysis for Public Safety, National Institute of Justice (2005)
12. Everitt, B.S., Landau, S., Leese, M.: Cluster Analysis, 4th edn. Oxford University Press, New York (2001)
13. Levine, N.: Risk-adjusted nearest neighbor hierarchical clustering, In: Levine, N. (ed.) CrimeStat III: A Spatial Statistics Program for the Analysis of Crime Incident Locations (version 3.0), Chapter 6. Ned Levine & Associates, Houston, TX and the National Institute of Justice, Washington, DC. November (2004)
14. Block, C.R.: STAC hot spot areas: a statistical tool for law enforcement decisions. In: Proceedings of the Workshop on Crime Analysis Through Computer Mapping. Criminal Justice Information Authority, Chicago, IL (1994)
15. Anselin, L.: Local indicators of spatial association – LISA. Geogr. Anal. **27**(2), 93–115 (1995)
16. Bailey, T.C., Gatrell, A.C.: Interactive Spatial Data Analysis. Longman Scientific & Technical, Burnt Mill, Essex, England (1995)
17. Brantingham, P.L., Brantingham, P.J.: Notes on the geometry of crime. In: Brantingham, P.J., Brantingham, P.L. (eds.) Environmental Criminology, pp. 27–54. Waveland Press, Inc., Prospect Heights, IL (1981)
18. Canter, D., Gregory, A.: Identifying the residential location of rapists, J. Forens. Sci. Soc. **34**(3), 169–175 (1994)
19. Rossmo, D.K.: Overview: multivariate spatial profiles as a tool in crime investigation. In: Block, C.R., Dabdoub, M., Fregly, S. (eds.) Crime Analysis Through Computer Mapping, pp. 65–97. Police Executive Research Forum: Washington, DC (1995)
20. Levine, N.: Bayesian Journey to Crime estimation (update chapter). In: Levine, N. (ed.) CrimeStat III: A Spatial Statistics Program for the Analysis of Crime Incident Locations (version 3.1). Ned Levine & Associates, Houston, TX and the National Institute of Justice, Washington, DC. March 2007. Available at http://www.icpsr.umich.edu/crimestat (2007)
21. Ortuzar, J.D., Willumsen, L.G.: Modeling Transport, 3rd edn. John Wiley & Sons, New York (2001)
22. Cameron, A.C., Trivedi, P.K.: Regression Analysis of Count Data. Cambridge University Press, Cambridge, UK (1998)
23. Domencich, T., McFadden, D.L.: Urban Travel Demand: A Behavioral Analysis. North-Holland Publishing Co., 1975. Reprinted 1996. Available at: http://emlab.berkeley.edu/users/mcfadden/travel.html (1975)
24. Sedgewick, R.: Algorithms in C++: Part 5 Graph Algorithms, 3rd edn. Addison-Wesley, Boston (2002)
25. Cohen, L.E., Felson, M.: Social change and crime rate trends: a routine activity approach, Am. Soc. Rev. **44**, 588–608 (1979)
26. Wilson, J.Q., Kelling, G.: Broken Windows: The Police and Neighborhood Safety. Atlantic Monthly **29**(3), 29–38 (1982)
27. Bibel, B.: Arrest Locations as a Means for Directing Resources. In: Levine, N. (ed.) CrimeStat III: A Spatial Statistics Program for the Analysis of Crime Incident Locations (version 3.0), Chapter 6. Ned Levine & Associates, Houston, TX and the National Institute of Justice, Washington, DC. November (2004)
28. Hill, B.: Catching the Bad Guy. In: Levine, N. (ed.) CrimeStat III: A Spatial Statistics Program for the Analysis of Crime Incident Locations (version 3.0), Chapter 10. Ned Levine & Associates, Houston, TX and the National Institute of Justice, Washington, DC. November (2004)
29. LeBeau, J.L.: Distance Analysis:Man With A Gun Calls For Servicein Charlotte, N.C., 1989. In: Levine, N. (ed.) CrimeStat III: A Spatial Statistics Program for the Analysis of Crime Incident Locations (version 3.0), Chapter 4. Ned Levine & Associates, Houston, TX and the National Institute of Justice, Washington, DC. November (2004)

30. Amaral, S., Monteiro, A.M.V., Câmara, G., Quintanilha, J.A.: Evolution of the Urbanization Process in the Brazilian Amazonia In: Levine, N. (ed.) CrimeStat III: A Spatial Statistics Program for the Analysis of Crime Incident Locations (version 3.0), Chapter 8. Ned Levine & Associates, Houston, TX and the National Institute of Justice, Washington, DC. November 2004 (2004)

Cross-Covariance Models

► Hurricane Wind Fields, Multivariate Modeling

CSCW

► Geocollaboration

Customization

► Mobile Usage and Adaptive Visualization

Cyberinfrastructure for Spatial Data Integration

ILYA ZASLAVSKY
San Diego Supercomputer Center, University of California San Diego, San Diego, CA, USA

Synonyms

E-Science; Heterogeneity; Virtualization, Resource; Standards

Definition

The term cyberinfrastructure (CI) refers to a new research environment that supports integration of geographically distributed computing and information processing services to enable a new level of data-intensive collaborative science enterprise. It includes high-performance data management and storage hardware and software, combined with secure data access and advanced information- and knowledge-management technologies and a variety of search, analysis, visualization, modeling and collaboration tools linked over high-speed networks, to create an enabling end-to-end framework for scientific discovery. CI applications in earth sciences span such disciplines as earthquake modeling and prediction, ecology, atmospheric sciences, hydrology and oceanography.

Historical Background

The term was first articulated at a press briefing on the Presidential Decision Directive (PDD-63) in 1998, in reference to information systems as the major component of the nation's critical infrastructures in need of protection (http://www.fas.org/irp/offdocs/pdd/pdd-63.htm). In 2003, the National Science Foundation's (NSF's) blue ribbon panel used the term in outlining the need to efficiently connect high-performance computing resources, information resources, and researchers, to support scientific discovery. Several large information technology projects were funded by the NSF, focused on CI development in earth science and other domains. In June 2005, an Office of Cyberinfrastructure was created at the NSF (http://www.nsf.gov/dir/index.jsp?org=OCI). At the same time, the National Institutes of Health supported advanced CI projects in biomedical sciences, and a range of infrastructure projects were developed in industry. These developments were accompanied by the emergence of relevant information exchange standards, more importantly *web services*, and *service-oriented architecture* (SOA), which now form the backbone of the large CI projects.

Several of these projects have been using spatial information technologies for monitoring distributed resources, searching for resources and extracting data fragments based on their spatial properties, integrating spatial data of different types, creating composite maps, and serving spatial data in standard formats, etc.

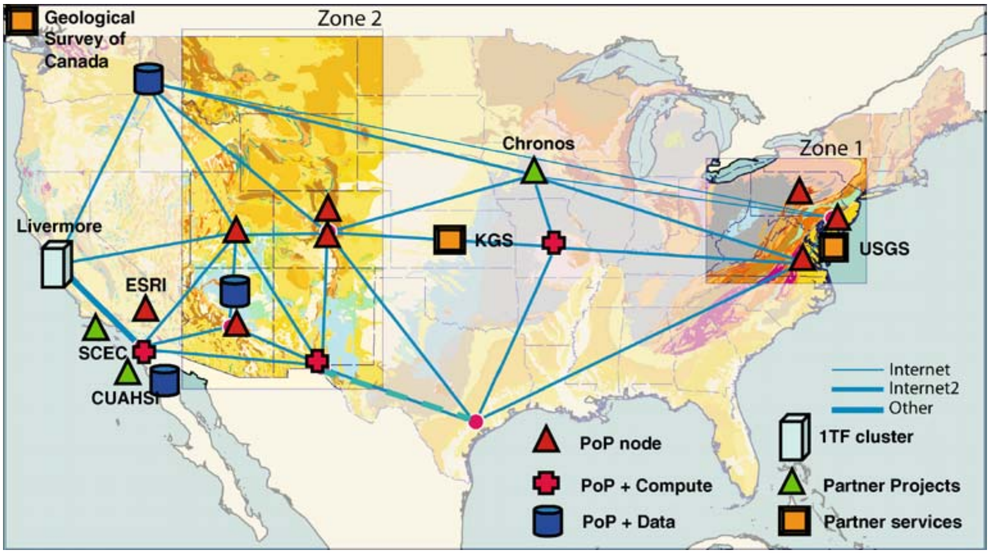
Scientific Fundamentals

The challenges of integrating spatial information from physically distributed spatial data sources derive from:

- Extreme heterogeneity in how web-accessible information is collected, represented, described, interpreted and queried
- Exponentially increasing volumes of available data, numbers of users and applications
- Volatility of the web, with autonomously managed data sources and services
- The need to transform data into standard agreed-upon forms suitable for spatial representation and analysis

The mechanisms proposed within CI projects to address these challenges follow the idea of *resource virtualization*, that is, decoupling information sources and services from their specific implementation, geographic location, or physical configuration. Such resources can be pooled together to address computation- or data-intensive problems. For spatial data, this typically involves:

- Standardization of spatial metadata, specifically following FGDC Content Standard for Digital Geospatial Metadata, ISO 19115 and 19139, and standards-compliant registration of different types of spatial data and services to catalogs that can be accessed and queries in a standard fashion



Cyberinfrastructure for Spatial Data Integration, Figure 1 Organization of the Geosciences Network (GEON) *grid*, showing locations of GEON point-of-presence (*PoP*) nodes, compute nodes, and data nodes, as well as partner institutions and services

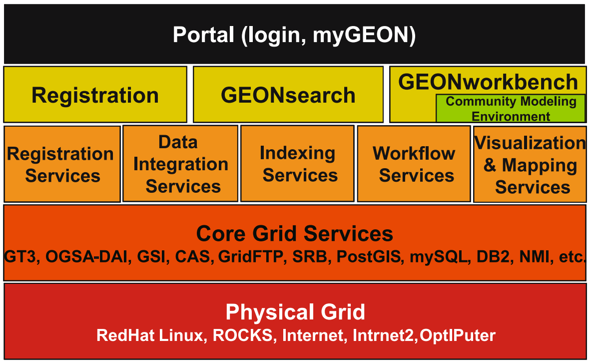
- Standardization of spatial data transmission on the web (GML (Geographic Markup Language), in particular), and XML-based (eXtensible Markup Language) standards for graphic rendering on the web (e. g., SVG (Scalable Vector Graphics))
- Wrapping of spatial data into standards-compliant GIS services, most notably following specification developed within the Open Geospatial Consortium
- Publication of geographic information systems (GIS) *web services* representing common GIS functionality which can be invoked from different client environments; and development of convenient platforms for generation of such services (e. g., ESRI's ArcGIS Server)
- Creation of advanced query-processing clients, in particular implementing *spatial information mediation* techniques for assembling spatial data fragments into composite query responses
- automatic map assembly services that support 'on-the-fly' generation of thematic maps from multiple grid-enabled data sources
- Single sign-on authentication, whereupon login users are issued a certificate that specifies their access rights with respect to different registered resources, which may have different security models
- Replication and caching of commonly used spatial data collections and metadata, to support quality of service and alleviate 'single point of failure' issues
- Development of spatial query processing architectures for distributing geospatial processing to multiple processing nodes (including the emerging P2P-type architectures for spatial data processing)
- Development of techniques for capturing data and service semantics, to enable semantic mediation across disparate sources using different vocabularies, and orchestration of services into processing pipelines (workflows)
- Development of principles for archiving large volumes of spatial data, and instantiating the state of GIS databases for a given time
- Development of map-based *grid* monitoring services.

Given the traditional GIS focus on spatial data infrastructure, standard description and reuse of secondary data sources, and support of spatial data integration, individual components of the CI agenda have been long explored in GIS literature. However, the experience of several large CI projects reviewed below and a series of interoperability initiatives led by the Open Geospatial Consortium, have demonstrated that these components can be integrated into functioning computing systems supporting discovery, integration and analysis of distributed spatial data resources and services.

Key Applications

Geology

The Geosciences Network (GEON) (www.geongrid.org) is a large multiuniversity project supported by the NSF (Fig. 1). Its mission is development of infrastructure for data integration in the geosciences. GEON's SOA includes



Cyberinfrastructure for Spatial Data Integration, Figure 2 GEON *grid* software layers. Software components spelled out: *ROCKS* SDSC clustering toolkit, www.rocksclusters.org; *OptIPuter* novel infrastructure integrating optical networking, internet protocol, computer storage, analysis and visualization, http://www.optiputer.net/; *GT3* GLOBUS Toolkit, v.3, www.globus.org; *OGSA-DAI* middleware supporting web service access to data resources, www.ogsadai.org.uk; *CAS* Community Authorization Service, http://www.globus.org/grid_software/security/cas.php; *GridFTP* (high-performance data transfer protocol, http://www.globus.org/grid_software/data/gridftp.php; *SRB* SDSC storage resource broker, http://www.sdsc.edu/srb/; *PostGIS* GIS extension to the PostgreSQL database management system, http://postgis.refractions.net/; *mySQL* open source database management system, www.mysql.com/; *DB2* IBM's DBMS, www.ibm.com/db2; *NMI* NSF middleware initiative tools, http://www.nsf-middleware.org/

tools for publication, search, analysis and integration of several types of spatial data (Fig. 2). GIS-related innovations in GEON include:

- Map-based *grid* node monitoring service
- Web-based environment for publishing and registering shapefiles, Web Map Service/Web Feature Service (WMS/WFS), grids, and other types of spatial data
- Ability to semantically annotate spatial data and services, and ontology-based search
- Map assembly services, for automatic generation of online thematic maps from fragments extracted from several registered data sources
- Ability to securely engage high-performance computing and networking resources (clusters, supercomputers) in computation- and data-intensive processing (e. g., with LIDAR (Light Detection And Ranging) datasets).

Ecology

The National Ecological Observatory Network (NEON) (www.neoninc.org) is designed as a nationwide measurement and observation system enabling regional- to continental-scale multidisciplinary ecological modeling and forecasting. While in its design phase at the time of writing, NEON is envisioned as an environment integrating sensor networks and advanced computing resources based on SOA and *grid* computing approaches.

Hydrology and Environmental Engineering

The Water And Environmental Research Systems (WATERS) network initiative includes two interrelated CI efforts supported by NSF: the Consortium of Universities for the Advancement of Hydrologic Sciences (CUAHSI) hydrologic information system (HIS) (www.cuahsi.org/his/), and the Collaborative Large-Scale Engineering Analysis Network for Environmental Research (CLEANER) (cleaner.ncsa.uiuc.edu) projects. The core of CUAHSI HIS, for example, is development of uniform data models and web service wrappers for heterogeneous repositories of observation data available from federal (e. g., United States Geological Survey (USGS), Environment Protection Agency (EPA), National Oceanic and Atmospheric Administration (NOAA), state, and local sources. Both projects have been actively using GIS for online browsing, search and visualization, and developed GIS-based client interfaces, both online and desktop, for the observation *web services* available from CUAHSI HIS.

Biomedical Sciences

The Biomedical Informatics Research Network (BIRN) (www.nbirn.net) project, supported by the National Institutes of Health, has pioneered many *grid* developments, by creating a production-level infrastructure for integrating neuroscience data across multiple participating universities. BIRN has a significant spatial information integration component, focused on combining information from multiple distributed atlases of animal and human brains. In particular, it supports spatial and semantic annotation of neuroscience images and segmentations, querying across distributed image collections registered to a common stereotaxic coordinate system and available through the BIRN data *grid*, and retrieving neuroscience images from GIS map services into GIS-based client interfaces. An extension of BIRN is the National Institute of Environmental Health Sciences (NIEHS) Hurricane Response Portal (http://www-apps.niehs.nih.gov/katrina/), where online GIS is integrated with portal technologies so that user access to various spatial data layers is controlled by credentials evaluated via the project's portal.

Other projects that focus on CI development within various earth science domains, and use GIS approaches, include the Ocean Research Interactive Observatory Networks (ORION) (http://www.orionprogram.org/), Real-Time Observatories, Applications, and Data Management

Network (ROADnet) (roadnet.ucsd.edu), Linked Environments for Atmospheric Discovery (LEAD) (lead.ou.edu). A range of similar efforts are supported by federal agencies such as USGS, EPA and NOAA, where relevant geographic data can be accessed via the Geospatial One Stop portal (GOS) (www.geodata.gov). Similar developments exist in Europe, for example, the UK e-Science program, with several GIS-focused efforts within the EDINA project, an on-line abstract and indexing database at Edinburgh University Library (http://edina.ac.uk/).

Future Directions

CI and services for spatial information integration is a rapidly developing area, expanding to new research domains that span from integration of archaeological data to plant sciences CI (judging from recently awarded or announced grant competitions). While each project focuses on a unique combination of domain and computer science research, several topics have emerged that will likely define CI development for several years to come. They include:

- Support for environmental observatories, and handling of heterogeneous streams of observation data
- Creation and management of large earth science ontologies, and ontology tagging of important data sets
- Efficient integration of multimodal and geographically distributed data sources
- Information integration across spatial scales
- Mining of web-accessible sources for place-related information and knowledge
- Efficient organization of distributed computational infrastructure for massive data-intensive computations, fault-tolerance and rapid response
- Spatiotemporal indexing and scheduling of grid resources
- Secure access to heterogeneous grid resources
- Social aspects of organizing geographically distributed research groups; and many more.

Cross References

► Grid
► Service-Oriented Architecture
► Spatial Information Mediation
► Web Services

Recommended Reading

Berman, F., Hey, A., Fox, G.: Grid Computing: Making The Global Infrastructure a Reality. Wiley, Indianapolis, IN, USA (2003)

Catlett, C., Smarr, L.: Metacomputing. Commun. ACM **35**:44–52 (1992)

Chervenak, A., Foster, I., Kesselman, C., Salisbury, C., Tuecke, S.: The data grid: towards an architecture for the distributed management and analysis of large scientific datasets. J. Network Comput. Appl. **23**, 187–200 (2001)

DeVogele, T., Parent, C., Spaccapietra, S.: On spatial database integration. Int. J. Geogr. Inf. Sci. **12**, 335–352 (1998)

Erl, T.: Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall, Upper Saddle River, USA (2005)

Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the grid: enabling scalable virtual organizations. Int. J. Supercomput. Appl. **15**:200–222 (2001)

Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The physiology of the grid: an open grid services architecture for distributed systems integration. Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002

Lacroix, Z., Boucelma, O., Essid, M.: A WFS-based mediation system for GIS interoperability. In: Workshop on Advances in Geographic Information Systems, pp. 23–28. ACM, New York (2002)

National Science Foundation (1993) NSF Blue Ribbon Panel on High Performance Computing, October 19, 1993 (NSB 93–205).Available via http://www.nsf.gov/pubs/stis1993/nsb93205/nsb93205.txt. Accessed 30 September 2005

O'Reilly, T. (2005) What is Web 2.0. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html. Accessed 2 November 2004

Reed, C. (2004) Data integration and interoperability: ISO/OGC standards for geo-information. www.directionsmag.com/article.php?article_id=687. Accessed

Zaslavsky, I., Baru, C., Bhatia, K., Memon, A., Velikhov, P., Veytser, V.: Grid-enabled mediation services for geospatial information. In: Agouris, P., Croitoru, A. (eds.) Next Generation Geospatial Information. From Digital Image Analysis to Spatiotemporal Databases, pp. 15–24. Taylor and Francis, London, UK (2005)

Data Acquisition

► Photogrammetric Applications

Data Acquisition, Automation

CHRISTIAN HEIPKE
Institute of Photogrammetry and GeoInformation,
Leibniz University Hannover, Hannover, Germany

Synonyms

Automatic information extraction; Image analysis; Scene analysis; Photogrammetry

Definition

Automatic data acquisition is the extraction of information from images, relevant for a given application, by means of a computer. Photogrammetric image processing is divided into two aspects, i. e., the *geometric/radiometric image evaluation* and *image analysis*. Geometric/radiometric image evaluation comprises image orientation, the derivation of geometric surface descriptions and orthoprojection. Image analysis contains the extraction and description of three-dimensional (3D) objects. A strict separation of both areas is possible neither for manual nor for automatic photogrammetric image processing.

Historical Background

In the past, geometric/radiometric image evaluation and image analysis were two clearly separated steps in the photogrammetric processing chain. Using analogue imagery, *automation* was understood as a supporting measure for a human operator, e. g., by driving the cursor automatically to a predefined position in image and/or object space to capture well-defined tie points or to speed up image coordinate measurement of ground control points or digital terrain model (DTM) posts. The first successful attempts towards a more elaborate role for the computer became commonplace once analogue images could be scanned and subsequently processed in digital form. In this way, interior and relative orientations, as well as large parts of aerial triangulation and DTM generation, became candidates for a fully automatic work flow. The recent development of digital aerial cameras inspires hope for further automation in the image analysis step.

Scientific Fundamentals

When using digitized or digitally acquired images, the border between geometric/radiometric image evaluation and image analysis becomes blurred, mostly because, due to automation, the formerly decisive manual measurement effort has lost much of its significance. Therefore, already in the orientation phase a point density can be used, which is sufficient for some digital surface models (DSMs). Methods for the integrated determination of image orientation, DSMs, and orthophotos have been known for some time, but for the sake of clarity the various steps shall be looked at separately here.

The components of image orientation are the sensor model, i. e., the mathematical transformation between image space and object space, and the determination of homologous image primitives (mostly image points). As far as the sensor model is concerned, the central projection as a classical standard case in photogrammetry must be distinguished from line geometry.

In the context of bundle adjustment the central projection is traditionally described by means of collinearity equations. It should be noted, however, that the resulting set of equations is nonlinear in the unknown parameters. Starting from these observations, and from the known problem of deriving initial values for image orientation, especially in close-range photogrammetry, alternative formulations for the central projection were examined, based on projective geometry [1]. If necessary, the results can be used as initial values in a subsequent bundle adjustment. Alternative linear methods are also in use with satellite images, where rational polynomials play a certain role.

The determination of homologue points is almost exclusively done by digital image matching. While in close-range photogrammetry this task is still a matter of active research due to the variable image perspectives and the large depth range, the methods for aerial images and for the satellite sector are almost fully developed and are available for practical purposes under the term "automatic aerial triangulation". It should be noted that the automatically generated image coordinates of the tie points are often interactively supplemented or corrected.

As an alternative to aerial triangulation the direct and integrated sensor orientation were thoroughly investigated in the last decade. In both cases data from global positioning system (GPS) receivers and inertial measurement units (IMUs) are used for determination of the elements of exterior orientation [2]. For direct sensor orientation these data replace tie and (more importantly) also ground control points and thus the entire aerial triangulation. For integrated sensor orientation, all information is used in a combined adjustment. In close-range photogrammetry, coded targets play a central role as ground control points, since their position in the images can be determined fully automatically.

Like image orientation the derivation of geometric surface descriptions from images is based on digital image matching. If a DTM is to be derived from the DSM, interfering objects (for the terrain these can be buildings, trees, etc.) must be recognized and eliminated. At present this task is solved by comparatively simple image processing operators and statistical methods. For aerial and satellite images DTM generation is commercially available in nearly every photogrammetry software package. As in image orientation, the automatic step is usually followed by a postediting phase to eliminate blunders and fill in areas in which matching was not successful. In close range, the problem of surface determination is different. Owing to smaller distances to the objects, more flexibility exists regarding the selection of sensors and evaluation method. Examples are the well-known coded light approaches and various so-called shape-from-X procedures, where X stands for motion, focus, contours, shading, and texture.

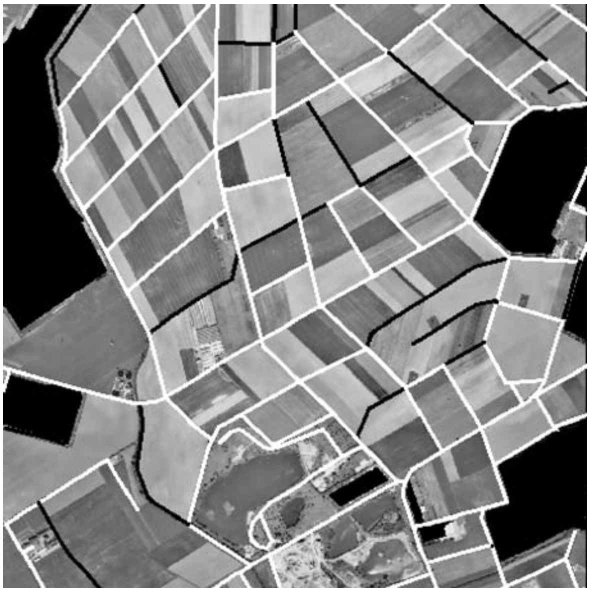
Orthoprojection, the projection of a central perspective image to a reference surface, mostly a horizontal plane, is a standard task in photogrammetry: Recently, automatic solutions for so-called true orthos have become available. True orthos are orthophotos for which a high quality DSM has been used for differential rectification, instead of a traditional DTM, and where occluded areas are filled in from neighboring images. As a result, for example, roofs and bridges are depicted at their geometrically correct position, and building walls are not visible.

Image analysis can be defined as the automatic derivation of an explicit and meaningful description of the object scene depicted in the images [3]. For this purpose, individual objects such as roads and buildings must be recognized and described. This recognition needs prior knowledge of objects in terms of models, which must be made available to the machine prior to starting the automatic process. Alternatively, they can also be learnt in a first step of the process itself. In order to set up useful models, geometric and radiometric information on the various objects must be collected and adequately represented. For aerial imagery, the larger the scale of the images to be analyzed and the more details are required, the more important is geometric information, as one increasingly enters into the domain of human activity, which can be characterized by linear borders, symmetries, right angles, and other geometric aspects. For smaller resolutions, however, radiometric and spectral attributes dominate, which explains the good results of multispectral classification for satellite images of coarser resolution, as well as the inferior results of the same technique for high-resolution satellite and aerial images.

The set-up of the object models is a major problem in image analysis. At present, despite significant research effort it is still not clear, a priori, which elements of an object and scene description need to be taken into account to build a useful model. Recently, more and more statistical methods are being used in knowledge acquisition and representation. Presently, these attempts are still provisional; however, it is obvious that an efficient automatic generation of models is a decisive prerequisite for image analysis to succeed altogether.

Another possibility for introducing a priori knowledge is based on the assumption that images are normally analyzed for a certain purpose, predefined at least in its main features. In geographical informational systems (GIS), for example, the available information is described in object catalogues, which contain relevant information for formulating the object models for image analysis. It is sometimes also postulated that object models for image analysis should be set up hierarchically, in a similar way as they are described in object catalogues: the upper level discerns only coarse context areas, such as settlements, forests, open landscape, and water bodies, and a refinement then follows within the respective context area.

Available GIS data rather than only descriptions in feature catalogues may also be used as part of the knowledge base. In this way, the GIS data can also be checked for correctness and completeness. An example is shown in Fig. 1, where road data are superimposed with an orthophoto. Roads depicted in white have been automatically checked and verified by the developed system; roads in black were



Data Acquisition, Automation, Figure 1 Orthophoto with superimposed road network from a geographical informational systems (GIS) database. Roads depicted in *white* were automatically detected in the orthophoto and could thus be verified, for roads in *black* this was not the case; the *black* roads need to checked by a human operator

not recognized automatically and need to be checked by a human operator [4]. The formal description of data quality is still an open, but important aspect for this approach.

In recent years, important progress has been made in image analysis, even though a breakthrough in the direction of practical applications has not yet been achieved. Under certain conditions single topographic objects like roads in open terrain, buildings and vegetation can be successfully extracted automatically. The present status of image analysis can be summarized as follows [5]:

- Simultaneous use of multiple images, combined with early transition to the 3D object space, simultaneous use of point, line and area information through projective geometry
- Rich modular object modeling encompassing geometric, radiometric, and spectral information
- Simultaneous use of multiple image resolutions and degrees of detail in object modeling in terms of multiscale analysis
- Simultaneous interpretation of different data sources, such as single images and image sequences with geometric surface descriptions and two dimensional maps;
- Modeling of context and complete scenes instead of single-object classes;
- Investigations regarding formulation and use of uncertain knowledge, for example based on graphical models such as Bayes nets, fuzzy logic, and evidence theory to enable automatic evaluation of the obtained results in terms of self-diagnosis;
- Investigations into automatic production of knowledge bases using machine learning

Key Applications

Automation in data acquisition from images finds a host of applications in all areas dealing with the determination of 3D coordinates as well as the interpretation of imagery. Traditionally, the key application of photogrammetry has been topographic and cartographic mapping. Recently, new technical developments such as digital still and video cameras and new demands such as environmental monitoring and disaster management have paved the way for many new applications.

Mapping and GIS still are the key applications today. Other disciplines such as agriculture, forestry, environmental studies, city and regional planning, 3D city modeling, geology, disaster management and homeland security also increasingly make use of automatic data acquisition from aerial and satellite images. In the close range, applications range from industrial metrology, location-based services (LBS), autonomous navigation and traffic monitoring, to architecture, archeology, cultural heritage and medicine.

Future Directions

In spite of a large body of successful research in recent years, practical applications of fully automatic systems do not seem realistic in the foreseeable future. Semiautomatic procedures, however, are beginning to be used successfully. Contrary to fully automatic methods, semiautomatic methods [6] integrate the human operator into the entire evaluating process. The operator mainly deals with tasks which require decisions (e. g., selection of algorithms and parameter control), quality control, and—where required—the correction of intermediate and final results.

It is anticipated that these semiautomatic approaches will be established in practical work within the next few years. A proper design of the man–machine interface will probably be of greater importance to the users than the degree of automation, provided that the latter allows for more efficiency than a solely manually oriented process.

Cross References

► Photogrammetric Methods
► Photogrammetric Sensors

Recommended Reading

Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)

D

Schwarz, K.-P., Chapman, M.E., Cannon, E., Gong, P.: An integrated INS/GPS approach to the georeferencing of remotely sensed data. Photogramm. Eng. Remote Sensing **59**, 1667–1674 (1993)

Rosenfeld, A.: Computer image analysis: an emerging technology in the service of society. Comput. Sci. Tech. Rep. TR-1177, MCS-79-23422, University of Maryland (1982)

Gerke, M., Butenuth, M., Heipke, C., Willrich, F.: Graph-supported verification of road databases. ISPRS J. Photogramm. Remote Sensing **58**, 152–165 (2004)

Baltsavias, E.: Object extraction and revision by image analysis using existing geodata and knowledge: current status and steps towards operational systems. ISPRS J. Photogramm. Remote Sensing **58**, 129–151 (2004)

Gülch E., Müller, H.: New application of semi-automatic building acquisition. In: Baltsavias, E., Grün, A., van Gool, L. (eds.) Automatic Extraction of Man-Made Objects from Aerial and Space Images (III), pp. 103–114. Balkema, Lisse (2001)

Data Analysis, Spatial

MICHAEL F. GOODCHILD
Department of Geography, University of California,
Santa Barbara, CA, USA

Synonyms

Spatial analysis; Geospatial analysis; Geographical analysis; Patterns; Anamolies; Spatial interaction; Point patterns; Geostatistics; GeoDa; Geographically weighted regression

Definition

Spatial data analysis refers to a set of techniques designed to find pattern, detect anomalies, or test hypotheses and theories, based on spatial data. More rigorously, a technique of analysis is spatial if and only if its results are not invariant under relocation of the objects of analysis—in other words, that location matters. The data that are subjected to spatial data analysis must record the locations of phenomena within some space, and very often that is the space of the Earth's surface and near-surface, in other words the geographic domain. However, many methods of spatial data analysis can prove useful in relation to other spaces; for example, there have been instances of methods of spatial data analysis being applied to the human brain or to the space of the human genome. The terms spatial data analysis, spatial analysis, and geographic analysis are often used interchangeably. Spatial data analysis overlaps very strongly with spatial data mining. Some authors use the latter term to refer specifically to the analysis of very large volumes of data, and to imply that the purpose is the detection of pattern and anomalies—in other words hypothesis generation—rather than the testing of any specific hypotheses or theories. In this sense spatial data mining is more strongly associated with inductive science than with deductive science. However to other authors the terms data analysis and data mining are essentially synonymous.

Historical Background

Modern interest in spatial data analysis dates from the 1960s, when the so-called quantitative revolution in geography was at its peak. A number of authors set about systematically collecting techniques that might be applied to the analysis of geographic data, in other words to patterns and phenomena on the Earth's surface, drawing from the literatures of statistics, geometry, and other sciences. Berry and Marble (1968) published one of the first collections, and included discussions of spatial sampling, the analysis of point patterns, the fitting of trend surfaces to sample data in space, measures of network connectivity, Monte Carlo simulation, and measures of spatial dependence. Other early texts were written by Haggett (1966), Haggett and Chorley (1969), King (1969), and Taylor (1977). The topic of spatial dependence quickly surfaced as one of the more unique aspects of geographic pattern, and Cliff and Ord (1973) unified and extended earlier work by Moran and Geary into a comprehensive treatment.

Many of these early efforts were driven by a desire to find general principles concerning the distribution of various types of phenomena on the Earth's surface. For example, Central Place Theory had postulated that under ideal geographic and economic conditions settlements on the Earth's surface should occur in a hexagonal pattern. Many methods of point pattern analysis were developed and applied in order to detect degrees of hexagonality, without success. Other researchers were interested in the morphological similarity of patterns across a wide range of phenomena, and the implications of such patterns for ideas about process. For example, Bunge (1966) describes efforts to compare the geometric shapes of meandering rivers with roads in mountainous areas, and others compared river and road networks to the geometry of branching in the human lung.

Another quite different direction might be described as normative, or concerned with the design and planning of systems on the Earth's surface. The field of location-allocation modeling developed in the 1960s as an effort to develop techniques for the optimal location of such central facilities as schools, fire stations, and retail stores (Ghosh and Rushton, 1987). Other researchers were concerned with the optimal design of voting districts or the optimal routing of power lines or roads across terrain.

Another large literature developed around the modeling of spatial interaction. The numbers of visitors to central facilities such as retail stores is observed to decline systematically with distance from home. Spatial interaction models attempt to predict such flows based on the characteristics of the home neighborhood, the characteristics of the destination, and the characteristics of the trip (Fotheringham and O'Kelly, 1989). They have been applied successfully to the modeling of migration, social interaction, and many other types of spatial interaction.

Interest in spatial data analysis has grown rapidly in recent years, in part because of the increasing availability of spatial data and the popular acceptance of tools such as Google Earth, Google Maps, and Microsoft Virtual Earth. Geographic information systems (GIS) are designed to support the manipulation of spatial data, and virtually all known methods of spatial data analysis are now available as functions within this environment. There has been some success at achieving interoperability between the many brands of GIS and formats of spatial data, so that today it is possible to submit spatial data to analysis in a uniform computing environment that is also equipped to perform the necessary ancillary tasks of data preparation, along with the visualization of results. Increasingly the results of spatial data analysis are portrayed through generic services such as Google Maps, which allow them to be "mashed" or combined with other information, allowing the user to explore the geographic context of results in detail.

Scientific Fundamentals

Statistical analysis evolved in domains where location was rarely important. For example, in analyzing the responses to a questionnaire it is rarely important to know where respondents live. It is possible in such situations to believe that the members of a sample were selected randomly and independently from some larger population. But when dealing with spatial data this assumption is rarely if ever true. The census tracts of Los Angeles, for example, clearly were not drawn randomly and independently from some larger set. Spatial data analysis must confront two tendencies that are almost always present, yet rarely present in other types of analysis: spatial dependence, or the tendency for local variation to be less than global variation; and spatial heterogeneity, or the tendency for conditions to vary over the surface of the Earth. Technically, these tendencies lead to an overestimation of the numbers of degrees of freedom in a test, and to an explicit dependence of results on the bounds of the test.

Faced with this reality, some texts on spatial data analysis have focused first on the normal assumptions of statistics, and then attempted to show how the reality of spatial data imposes itself. It is in many ways more satisfactory, however, to proceed in reverse—to first discuss the spatial case as the norm, and then to introduce the assumptions of independence and homogeneity.

It is helpful to define spatial data rigorously, since the definition of spatial data analysis depends on it. Data may be defined as spatial if they can be decomposed into pairs of the form **<x, z>** where **x** denotes a point in space-time and **z** denotes one or more properties of that point. It is common to distinguish between spatial or geographic analysis, conducted in two or three spatial dimensions, and spatio-temporal analysis in which the temporal dimension is also fundamental; thus spatial data analysis may involve two, three, or four dimensions.

This atomic form of spatial data is rarely observed, however, because in principle an infinite number of points can be identified in any geographic domain—only in the case of data sampled at a finite number of points is this form actually analyzed. In other cases spatial data consist of aggregate statements about entire lines, areas, or volumes. For example, summary census data are statements about entire counties, tracts, or blocks, since issues of confidentiality prohibit publication of data about individuals. Moreover, data about interactions are statements about pairs of such objects; for example, a state-to-state migration table contains 2500 entries, each giving the number of migrants between a pair of states. The rich variety of forms of aggregation that are used to publish spatial data lends complexity to the field, and has led many authors to organize surveys on this basis.

For example, Bailey and Gatrell (1995) organized their text into four major sections based on data type. Patterns of undifferentiated points were the basis for the first, and techniques are described for estimating a surface of point density, for comparing patterns of points to a statistical model of randomness, and for detecting clusters in spatial and spatio-temporal point patterns. Such methods are widely employed in the analysis of patterns of disease and in biogeography. The second major section also focuses on points, but as samples of continuous phenomena that are conceptualized as fields. Geostatistics provides the theoretical basis for many of these techniques, since one of the most popular tasks is the interpolation of a complete surface from such sample point data. The third major section concerns areal data, typified by the aggregate statistics reported by many government agencies. Such data are widely used to estimate multivariate models, in the analysis of data on crime, economic performance, social deprivation, and many other phenomena. Several specialized techniques have been developed for this domain, including various forms of regression that are adapted to the special circumstances of spatial data. Finally, the last major section is devoted to the analysis of spatial interaction data and to various forms of spatial interaction modeling.

The widespread adoption of GIS has had profound effects on all aspects of spatial data analysis. Several authors have discussed this relationship, and texts that have appeared in the past decade, such as that by O'Sullivan and Unwin (2003), are clearly informed by the theories and principles of geographic information science (GIScience). Recent texts on GIS (e. g., Longley et al., 2005) also place spatial data analysis within this increasingly rigorous framework.

GIScience draws a clear distinction between two alternative conceptualizations of space: as a set of continuous fields, and as a collection of discrete objects occupying an otherwise empty space. The field/object dichotomy is clearly evident in the work of Bailey and Gatrell (1995), but becomes explicit in more recent texts. Continuous fields must be discretized if they are to be represented in digital systems, in one of a number of ways. In principle one would like the methods and results of spatial data analysis to be independent of the method of discretization used, but in practice each method of discretization has its own methods of analysis, and much effort must be expended in converting between them. The most convenient discretization is the raster, in which fields are represented as values of a regular square grid, and Tomlin (1990) and others have shown how it is possible to achieve a high level of organization of the methods of spatial analysis if this discretization is adopted. The isoline discretization, in which a field is represented as a collection of digitized isolines, is far less convenient and comparatively few methods have been developed for it. The irregular sample point discretization has already been discussed in the context of geostatistics.

Longley et al. (2005) adopt a quite different way of organizing spatial data analysis, based on a hierarchy of conceptual complexity, and within the context of GIS. The simplest type in their scheme consists of query, in which the analyst exploits the ability of the GIS to present data in different views. This is in large part the basis of exploratory spatial data analysis, a subfield that provides the user with multiple views of the same data as a way of gaining additional insight. For example, the multiple views of a spatial data set might include a map, a table, a histogram, or a scatterplot. Anselin's GeoDa (geoda.uiuc.edu) is a current example of this style of computing environment, and supports many other types of view that are designed to expose potentially interesting aspects of data. Indeed, GIS has been defined as a system for exposing what is otherwise invisible in spatial data. In GeoDa views are dynamically linked, so that a user-defined selection in the map window is automatically highlighted in all other open windows.

Longley et al.'s second type is measurement, since much of the motivation for the original development of GIS stemmed from the difficulty of making manual measurements of such spatial properties as length, area, shape, and slope from maps. The third is transformation, and occurs whenever spatial data analysis results in the creation of new views, properties, or objects. For example, density estimation results in the creation of a new continuous field of density from a collection of discrete points, lines, or areas, and spatial interpolation results in the creation of a new continuous field from point measurements.

The fourth is descriptive summary, or the calculation of summary statistics from spatial data. A vast number of such measures have been described, ranging from the spatial equivalents of the univariate statistics (mean, median, standard deviation, etc.) to measures of fragmentation and spatial dependence.

Recently several new methods have been described that disaggregate such measures to local areas, reflecting the endemic nature of spatial heterogeneity. Such place-based methods are typified by the local Moran statistic, which measures spatial dependence on a local basis, allowing the researcher to see its variation over space, and by Geographically Weighted Regression (Fotheringham, Brunsdon, and Charlton, 2002), which allows the parameters of a regression analysis to vary spatially.

Longley et al.'s fifth category is design, or the application of normative methods to geographic data, and includes the optimization methods discussed earlier. The sixth, and in many ways the most difficult conceptually, is statistical inference, addressing the problems discussed earlier that can render the assumptions of normal statistical inference invalid. Several methods have been devised to get around these assumptions, including tests based on randomization, resampling so that observations are placed sufficiently far apart to remove spatial dependence, and the explicit recognition of spatial effects in any model.

Key Applications

Spatial data analysis is now commonly employed in many areas of the social and environmental sciences. It is perhaps commonest in the sciences that employ an inductive rather than a deductive approach, in other words where theory is comparatively sparse and data sets exist that can be explored in search of patterns, anomalies, and hypotheses. In that regard there is much interest in the use of spatial data analysis in public health, particularly in epidemiology, in the tradition of the well-known work of Snow on cholera (Johnson, 2006). Mapping and spatial data analysis are also widely employed in criminology, archaeology, political science, and many other fields. Goodchild and Janelle (2004) have assembled a collection of some of the best work from across the social sciences, while comparable collections can be found in ecology, environmental science, and related fields.

Spatial data analysis is also widely employed in the private sector and in administration. It is used, for example, by political parties to analyze voting behavior; by insurance companies to measure the geographic distribution of risk; and by marketing companies in organizing direct-mail campaigns and in planning retail expansions. The field of geodemographics focuses on the use of spatial data analysis to create and use detailed information on social, economic, and purchasing patterns in support of retailing and other forms of commercial activity.

Future Directions

Spatial data analysis provides a particular kind of lens for viewing the world, emphasizing the cross-sectional analysis of snapshots rather than the longitudinal analysis of changes through time, or approaches that ignore both the spatial and temporal dimensions and their power in organizing information and providing context. This is a time of unprecedented opportunity for spatial data analysis, for a number of reasons. First, spatial data and the tools needed to support spatial data analysis have evolved very rapidly over the past decade or so, and researchers are now able to perform a wide variety of powerful forms of analysis with considerable ease. Second, and not unrelated to the first point, interest in a spatial perspective has grown rapidly in recent years, and there have been many comments on the appearance of a spatial turn in many disciplines. Within this context, spatial data analysis is part of a much larger interest in space, that extends from tools and data to theory, and might be summarized under the heading of spatial thinking. The widespread availability of sophisticated tools such as Google Earth (earth.google.com) has drawn attention to the need for education in the basic principles of a spatial approach.

The past decade has witnessed a fundamental shift in the nature of computing, and in how it is used to support research and many other forms of human activity. Many of the tools of spatial data analysis are now available as Web services, obviating the need for individual researchers to acquire and install elaborate software and data. Many agencies now offer primitive forms of spatial data analysis over the Web, allowing users to map, query, and analyze the agency's data in a simple, easy-to-use environment and requiring nothing more than a Web browser. Large software packages are now typically constructed from reusable components, allowing the functions of different packages to be combined in ways that were previously impossible, provided each is compliant with industry standards.

Spatial data analysis is now evolving into much stronger support of the spatio-temporal case, through the construction of packages such as Rey's STARS (stars-py.sourcforge.net), and through the development of new and powerful techniques. In this arena much remains to be done, however, and the next decade should see a rapid growth in spatio-temporal techniques and tools.

Cross References

► Spatial Econometric Models, Prediction
► Statistical Descriptions of Spatial Patterns

Recommended Reading

Bailey, T.C., Gatrell, A.C.: Interactive Spatial Data Analysis. Longman Scientific and Technical, Harlow (1995)
Berry, B.J.L., Marble, D.F. (eds.): Spatial Analysis: A Reader in Statistical Geography. Prentice Hall, Englewood Cliffs, NJ (1968)
Bunge, W.: Theoretical Geography. Gleerup, Lund (1966)
Cliff, A.D., Ord, J.K.: Spatial Autocorrelation. Pion, London (1973)
Fotheringham, A.S., Brunsdon, C., Charlton, M.: Geographically Weighted Regression: The Analysis of Spatially Varying Relationships. Wiley, Hoboken, NJ (2002)
Fotheringham, A.S., O'Kelly, M.E.: Spatial Interaction Models: Formulations and Applications. Kluwer, Boston (1989)
Ghosh, A., Rushton, G. (eds.): Spatial Analysis and Location-Allocation Models. Van Nostrand Reinhold, New York (1987)
Goodchild, M.F., Janelle, D.G.: Spatially Integrated Social Science. Oxford University Press, New York (2004)
Haggett, P.: Locational Analysis in Human Geography. St Martin's Press, New York (1966)
Haggett, P., Chorley, R.J.: Network Analysis in Geography. Edward Arnold, London (1969)
Johnson, S.: The Ghost Map: The Story of London's Most Terrifying Epidemic—and How It Changed Science, Cities, and the Modern World. Riverhead, New York (2006)
King, L.J.: Statistical Analysis in Geography. Prentice Hall, Englewood Cliffs, NJ (1969)
Longley, P.A., Goodchild, M.F., Maguire, D.J., Rhind, D.W.: Geographic Information Systems and Science. 2nd edn. Wiley, New York (2005)
O'Sullivan, D., Unwin, D.J.: Geographic Information Analysis. Wiley, Hoboken, NJ (2003)
Taylor, P.J.: Quantitative Methods in Geography: An Introduction to Spatial Analysis. Houghton Mifflin, Boston (1977)
Tomlin, C.D.: Geographic Information Systems and Cartographic Modeling. Prentice Hall, Englewood Cliffs, NJ (1990)

Data Approximation

► Constraint Databases and Data Interpolation

Data Collection, Reliable Real-Time

VANA KALOGERAKI[1], AMIR SOHEILI[2]
[1] Dept. of Computer Science and Engineering, University of CA, Riverside, CA, USA
[2] ESRI, Redlands, CA, USA

Synonyms

Spatial queries; Reliable real-time data collection; Monitoring; Surveillance; Energy-aware; Energy optimization; Peer-tree (spatial index); Routing; SPIX; Spatial index

Definition

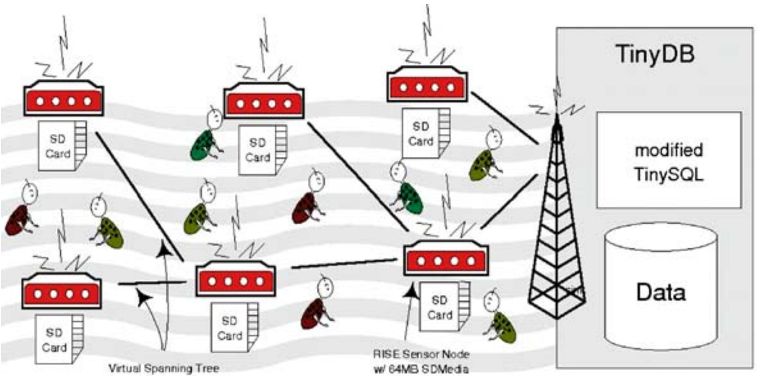
Recent technological advances in wireless technologies and microelectronics have led to the emergence of wireless sensor networks, consisting of large numbers of small, low-power, inexpensive wireless sensor devices that are embedded in the physical world and are able to monitor it in a non-intrusive manner. Wireless sensor networks have created tremendous opportunities for a wide variety of application settings. Large-scale wireless sensor network deployments have emerged in environmental and habitat monitoring, agriculture, health care, homeland security and disaster recovery missions. Different from general computer systems, wireless sensor devices are significantly constrained in terms of processing power, communication capability and energy. Furthermore, sensors are prone to failures due to manufacturing defects, environmental conditions or batter depletion. In such cases, the data may become stale or get lost. Failed sensors may introduce inconsistencies when answering queries and thus must be replaced to repair the network.

One of the most important operations in a sensor network is the ability to collect spatial data. *Spatial queries* are a subset of queries in which the database or the sensor network is queried by location rather than an attribute. The ability to collect spatial data is extremely useful for sensor networks in which sensors are deployed to gather physical data or monitor physical phenomena. Figure 1 shows an example of spatial data collection in a sensor network. The typical application is to monitor a geographical region over a time period and to collect all the data in this time window. Sensor nodes collect data and transmit them, possibly compressed and/or aggregated with those of the neighboring nodes to other nodes or to a central server (i. e., sink). Spatial queries are used to answer questions, such as *find the average temperature in an area or count the number of sensors within one mile of a point of interest*.

Processing spatial queries in traditional databases differ from sensor networks. The unique characteristics of the sensor devices generate new challenges for processing spatial queries in sensor network settings:

1. Distributed query execution. Queries must run in a distributed manner, because sensor data is distributed in the network and there is no global collection of the data in any of the nodes.
2. Distributed and dynamic index maintenance. The high energy cost of communication requires an efficient way to decide where to run the query to optimize the energy usage.
3. Reliable execution. To deal with sensor failures, the index must be able to reconstruct itself to minimize data inconsistencies and allow the correct sensors to continue execution. This mechanism will refine the structure of the index to respond to failures and save energy in spatial query processing.

To address these problems, one needs: (i) a distributed spatial index structure over the sensor network, maintained by the sensor nodes, to process spatial queries in a distributed fashion, and (ii) a distributed way of constructing, maintaining and optimizing the distributed spatial index to efficiently and reliably process the spatial queries while reducing energy consumption.



Data Collection, Reliable Real-Time, Figure 1 Example of Spatial Data Collection

Historical Background

Spatial query processing has been studied extensively in centralized systems. In a traditional spatial database, spatial indexing techniques such as R-Tree, R^+-Tree, and R^*-Tree [1,5,10] are used to execute a spatial query. R-Tree [5] is one of the most popular spatial index structures that has been proposed. In R-Tree each spatial data object is represented by a Minimum Bounding Rectangle which is used to store the leaf node entries in the form of (*Ptr*, *rect*) where *Ptr* is a pointer to the object in the database and *rect* is the MBR of the object. Non-leaf nodes store an MBR that covers all the MBRs in the children nodes. Variants of the R-Tree structure such as R+Tree [10] and R*Tree [1] have also been proposed. In a sensor network, however, spatial queries have to be processed in a distributed manner. Due to energy and computing power limitations of sensor nodes, computationally sophisticated approaches like the R-Tree or its distributed variants are not directly applicable to the sensor networks environment. In a sensor network, it is desirable to process the query only on those sensors that have relevant data or are used to route the data to the base station.

Range queries have also been studied in dynamic and large-scale environments [12]. However because of the resource limitation of the sensors, building a centralized index, a distributed index or a super-peer network to facilitate executing queries is not practical in a sensor network. Ferhatosmanoglu et al. [3] have proposed peer-tree, a distributed R-Tree method using peer-to-peer techniques in which they partition the sensor network into hierarchical rectangle shaped clusters. Similar to R-Tree, their techniques implement joins/splits of clusters when the number of items (sensor nodes) in the cluster satisfies certain criteria. The authors have shown how to use the peer-tree structure to answer Nearest Neighbor queries. In [3], the peer-tree is created bottom up by grouping together nodes that are close to each other. Each group of nodes selects a representative, which acts as the parent of this group of nodes, and these representatives are in turn grouped together at the next level. As a result, the connections between parents and children become progressively longer, and there is no way to guarantee that they can be implemented as single hops in the sensor network unless the assumption is made that the nodes' transition range is in the order of the dimensions of the sensor field. It is noted, however, that such long transmission ranges would have large energy costs. This technique, on the other hand, operates in a top down fashion when constructing the hierarchy, and guarantees that each parent to child connection is only one hop away.

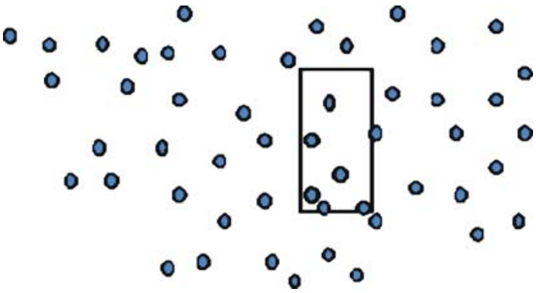
Other techniques have been proposed to reduce energy usage in sensor networks. LEACH [6] proposes an energy adaptive efficient clustering to distribute energy load evenly among the sensors in the network. Hu et al. [7] present a proactive caching scheme for mobile environment. Their idea is to create an index (R-Tree) and a cache from the spatial query results and use the cached records to reduce the size of the subsequent spatial query area. The cache and the index are stored in the mobile base station. These techniques reduce the size of the queried area and the number of requests that need to be sent to the sensor network, thus saving energy and increasing the lifetime of the sensor network. Unlike the above approaches, the spatial index is designed in a way that it can be applied to sensor networks with limited resources for processing spatial queries.

Many routing protocols have been proposed to route a packet to a specific location in the sensor network. Direct Diffusion [4] forwards the request based on the sender's interest such as location. Geographic based routing [8] use geographic coordinates to route queries to a specific sensor. Unlike the general routing protocols, the focus is on running spatial queries in a specific area of the sensor network. This approach builds a distributed spatial index over the sensor network which at the same time reduces energy consumption in disseminating and processing spatial queries.

Attribute-based query processors have also been developed for sensor networks. Systems like Cougar [2] and TinyDB [9] process attribute queries using a declarative SQL language. Their goal is to use sensor resources in an efficient way when collecting the query results. Madden et al. [9] have proposed an Acquisitional Query Processor (ACQP) that executes attribute queries over a sensor network. ACQP builds a semantic routing tree (SRT) that is conceptually an attribute index on the network. It stores a single one-dimensional interval representing the range values beneath each of the node's children. Every time a query arrives at a node, the node checks to see if any of its children values overlap with the query range. If so, it processes and forwards the query. SRT provides an efficient way for disseminating queries and collecting query results over constant attributes. Although collecting spatial query results in a spatially enabled sensor network is the same as collecting attribute query results, it is possible to significantly reduce the energy consumption in processing spatial queries by exploiting the fact that the sensors with the query result are usually located in the same geographical area.

Scientific Fundamentals

First, the system model is described. Consider a set of n sensor nodes deployed in a geographical area of interest. Each sensor i has a sensing radius i_s and a communica-



Data Collection, Reliable Real-Time, Figure 2 A range query example: the use is interested only in the values of the sensors faling in the query rectangle

tion radius i_c. In the sensor network, assume that the sensors are static and aware of their location. Assume that the sensors may fail from the network at any time. It is also assumed that sensors in the sensor network may be heterogeneous in transmission and processing power, but they use the same energy when processing a specific task or transmitting a radio signal. As in the attribute-based sensor network query processors [2,9], each sensor maintains data as a single table with two columns for the sensor geography (X and Y location). Queries are parsed and disseminated into the sensor network at the base station and a spatial query over a sensor network can return a set of attributes or an aggregation of attributes of sensors in any area of the sensor network.

Definition: Spatial Query: A spatial query in a sensor network S with n sensors is a function $F\{v_i | s_i \in Q\}$, in which $v_i \in R$ is the value of sensor i and $s_i \in R^2$ is its location (the values are real numbers and the locations are x, y coordinates). Function F can be an aggregate, such as SUM, MAX, MIN, AVG, applied to a set of values, and Q is a range of the form $[a, b] \times [c, d]$, ($a, b, c, d \in R$, that is, a, b, c, d, are real numbers, $a < b$, $c < d$); a sensor is in the area when its x coordinate is between a and b and its y coordinate is between c and d.

Figure 2 gives an example of a spatial query on a sensor network. Although the main interest is in techniques to efficiently evaluate aggregates such as SUM, MAX or MIN, the techniques described are general techniques tare general and can be used for other hierarchically decomposable functions. Alternative ways to define ranges (such as the intersection of arbitrarily oriented halfspaces) are also possible. This allows finding and/or aggregating attributes of sensors located within a defined area of interest such as a window, circle, polygon or trace. A spatial query has one or more spatial constraint which represents the area of interest. Let q be the area of interest. Sensor s located at position p satisfies the spatial query constraint if p is inside q.

The idea behind spatial data collection is to create a spatial index on groups of objects which are geographically related, and use this index to process spatial queries. Spatial queries are used to answer questions such as "*what is the average temperature in the region R?*". Spatial query processors typically execute spatial queries in two steps; a coarse grained search to find sensors in the minimum bounding rectangle of the area of interest and a fine grained search to filter out sensors that do not satisfy the spatial constraint. Therefore, unlike traditional attribute queries, spatial queries require that the sensor network understands more complex data types like points and polygons. Operations on these types are more complex when compared to operations on simple types.

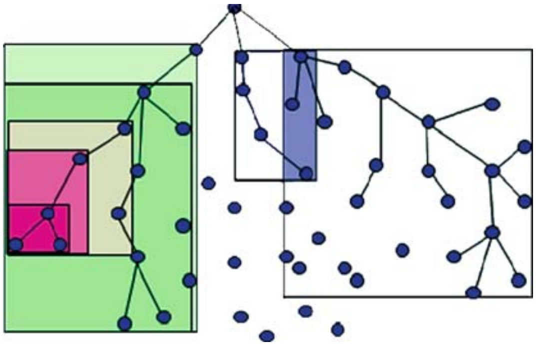
In a spatial database, a spatial index [1,5] will be used to identify nodes that intersect the minimum bounding rectangle (MBR) of the area of interest. These nodes will then be filtered out if they do not satisfy the spatial constraint. In a sensor network, sensors may join or fail at any time and the base station may not be aware of the location of all sensors at all times, so the base station may not be able to create a complete spatial index of the currently active sensors in the network.

The technique described below is a decentralized approach of executing spatial queries in a sensor network. The technique makes no assumptions about the capabilities of the sensor node while providing a protocol that reduces the network energy consumption in processing spatial queries.

Spix Index Structure. In this section SPIX [11] is described, a distributed index structure that allows each sensor to efficiently determine if it needs to participate in a given spatial query. SPIX is an index structure built on top of a sensor network, which essentially forms a routing tree that is optimized for processing spatial queries in sensor networks, as shown in Fig. 2. The spatial query processor running on each sensor uses this index structure to:

1. Bound the branches that do not lead to any result.
2. Find a path to the sensors that might have a result.
3. Aggregate the data in the sensor network to reduce the number of packets transferred and save energy.

SPIX imposes a hierarchical structure in the network. It makes the assumption that spatial queries will always be disseminated into the sensor network from a base station. The base station is responsible for preparing the query, submitting it into the sensor network and getting the result back. The spatial query will be disseminated into the routing tree and the result will be sent back to the root (base station). When a sensor receives the query, it must decide if the query applies locally and/or needs to be submitted to one of its children in the routing tree. A query applies locally if there is a non-zero probability that the sensor produces a result for the query.



Data Collection, Reliable Real-Time, Figure 3 A SPIX Index Structure example

Each sensor node in SPIX maintains a minimum bounded area (MBA), which covers itself and the nodes below it. Figure 3 shows an example of the tree built by SPIX, and some of the MBAs routed at some nodes. Each of these MBAs covers the subtree routed at the corresponding node. When a node receives a spatial query, it intersects the query area to its MBA. If the intersection is not empty, it applies the query and forwards the request to its children. Clearly, sensors with smaller MBAs have a higher chance to determine if the query applies to them and/or the sensors below them accurately and therefore save more energy in processing spatial queries.

SPIX exploits two models for creating a routing tree, Rectangle Model and Angular Model. In the rectangular model, the MBA is the minimum bounded rectangle (MBR) that covers the node and all nodes below it. In the Angular model, the MBA is the minimum bounded pie represented by start/end radius and start/end angles. The goal is to minimize the MBA area and MBA perimeter in SPIX to reduce energy consumption in the sensor network. Angular model is more effective when the base station queries the sensor network based on the distance between base station and the sensors. Rectangular model is more effective in other cases.

Building SPIX. Building SPIX is a two phase process:

1. *Advertisement phase*: The advertisement phase starts from the base station. In the advertisement phase, each sensor waits to receive an advertisement before it advertises itself to the sensors in its transmission range. The advertisement includes the location of the base station and the advertiser. The sensors maintain a list of advertisements they have received for the parent selection phase. The advertisement phase continues until all the sensors in the network hear an advertisement.
2. *Parent selection phase*: If a sensor has no children, it chooses its parent. If a sensor has candidate children, it waits until they select their parent and then it starts the parent selection phase. This phase can also be started when a timer expires. The closer the sensor is to the base station, the longer it needs to wait to start this phase. The parent selection phase continues until all the sensors in the network select their parent.

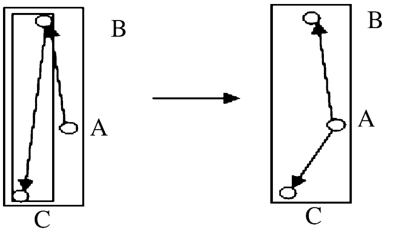
In order to avoid disconnections or cycles in the network, the base station submits its location to the sensors in the advertisement phase. Each sensor reviews its candidate parents before starting the parent selection phase and if there is at least one candidate closer from this sensor to the base station, it removes all the candidates that are farther from this sensor to the base station from the candidate parent list. In the maintenance phase, these candidates will be re-considered.

During the waiting period, when a sensor hears a parent selection message from another sensor, it updates its MBA, adds the sensor to its children list and notifies its vicinity that its MBA is updated. The *vicinity* of a sensor s is defined as being the set of sensors that are one hop away from s. Sensors in its vicinity are one hop away from it and thus the notification can be sent by broadcasting a message to its vicinity. When a sensor notices that its children's MBA is updated, it updates its MBA and notifies its vicinity.

Parent Selection Criterion. The parent selection criterion is important because the structure of the routing tree determines the way that the query is propagated in the sensor network and the efficiency of the spatial query execution. A weak parent selection criterion might create long and thin rectangles, which increases the transmission range, or increases the overlapped area dramatically and as a result queries more sensors during processing. Based on the experiences with R-Tree and its variants, choose two criteria for selecting the parent based on area and perimeter enlargement, and evaluate them in polar and coordinate systems. When a sensor wants to select a parent, it chooses a parent whose MBA needs the least area or perimeter enlargement to include the MBA of this sensor. If it finds two or more parent with the same area/perimeter enlargement, it chooses the parent that is geographically closer. Minimizing MBA perimeter enlargement would create more square-like rectangles and prevents creating long and thin rectangles [1].

Eliminating Thin Rectangles. Each sensor selects its parent based on the increase in parent MBA area or perimeter. This criterion might create long range radio communication links which is not desirable. In order to eliminate thin rectangles, the links are optimized as below:

When a sensor selects its parent, it notifies it children. When a child notices that it is closer to its grandparent than its parent, it disconnects from its parent and selects the grandparent as the new parent. This method eliminates large and thin rectangles, which is necessary when sensors



Data Collection, Reliable Real-Time, Figure 4 Effect of eliminating thin rectangles

are not close, but their X or Y coordinates is the same or is close enough to make thin rectangles. Figure 4 shows the sensor connections before and after the optimization.

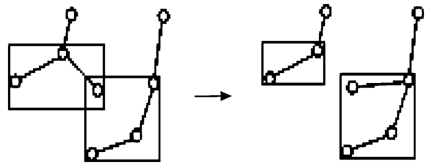
Energy Optimization Phase. In the energy optimization phase, the sensor network tries to reduce MBA areas of its sensors by re-organizing the sensor network leaf sensors. This would reduce the MBA area of the higher level sensors significantly when the sensor joins another branch. Sensors with smaller MBA have a higher chance of determining if the query applies to them and/or the sensors below them accurately and therefore saves more energy in processing spatial queries. When a spatial query propagates in the sensor network, each sensor intersects its MBA with the queried area. If a sensor finds that the intersection area is zero, it knows that the query does not apply to it and its children and therefore does not propagate it further. It is worth pointing out that sensors with a smaller MBA area have a higher chance to say "no" to a query and save energy.

When a leaf sensor determines that it is located in another sensor MBA area, it runs "parent-switching verification" process. It asks its parent: "What would be your MBA if I leave you?" If the parent's determines that its MBR would be zero without this child, it forwards the question to its parent. The question will not propagate further and the answer will be forwarded to the leaf node. If the leaf node determines that it is not located in the replied MBR, it disconnects from its parent and runs the join process.

In the energy optimization phase, the sensor network moves some sensors from one branch to another to reduce the MBR area of the higher level sensors. This transfer reduces the overlapped area of the sensors and therefore saves energy. Since the MBR size is reduced, fewer numbers of sensors will be involved in query processing and the system responds faster. Fig. 5 shows how the energy optimization phase reduces the MBR size of the higher level sensors.

Maintenance Phase. During the maintenance phase, sensor nodes may fail or new sensors may be added to the network. This may happen in the following conditions:

1. One or more sensors are added to the network.



Data Collection, Reliable Real-Time, Figure 5 Effect of Energy Optimization Phase

2. A sensor fails to respond and thus it must be removed from the SPIX structure.
3. A sensor did not join the network during the building phase.

a) Sensor Node Joins

To add a new sensor node in the network the Join phase is executed. In the Join phase, a sensor s broadcasts a message to its vicinity and requests for advertisement. Sensor nodes that hear the request for advertisement and are connected to the tree, send back their location and MBA to the sensor s. When sensor s receives a reply, it adds the sender node to the candidate parent list. When it hears from all sensors in its vicinity, it selects the parent as follows.

The criterion for choosing a parent among all candidates is to choose the parent which results in the minimum increase in its MBA area to include that sensor. If most of the candidate parents have small MBA area (or possibly zero area), then the criterion becomes choosing the parent that is geographically closer. Joining the closest parent may cause a large increase in the grandparent MBA size. Since the grandparent has at least one child, there is no need to check more than 2 hops away. Therefore, during the maintenance phase when a sensor determines that most of its candidate parents have zero or very small MBAs, it requests for a two hops parent selection. Each candidate parent replies with the MBA of its parents and the node chooses the parent that satisfies the two-hope parent selection criterion the best.

b) Sensor Node Failures

In order to determine sensor node failures, the "soft-state" stabilization technique is used. A lease (timeout period) is assigned on the children. When the lease expires, the sensor verifies the correctness of the parent-child relationships and recalculates its MBR, if necessary. Similarly, every time a sensor hears from its parent, it sets a random timer (greater than the lease period). When this timer expires, the child initiates the parent-children verification process.

Each sensor stores a list of its children and their MBRs. Sensors may fail at any time; when a child does not respond to a query or the lease expires, the sensor determines that its child has failed and it re-computes its MBR using the stored values. When the MBR size changes, the

sensor notifies the sensors in its vicinity. MBR updates may be propagated all the way to the base station.
Recalculating the new parent produces extra overhead in the sensor network. The sensor must select its parent, which requires sending several messages to the sensors in its transmission range and after selecting the parent, the MBR of the branch connecting the sensor to the base station needs to be updated. To reduce the number of messages, MBR updates will be propagated only when a random timer expires.
When a sensor fails, its children become orphans. Orphan nodes run a "Join" process to join the network again. Because of the limited communication range, it is possible that a sensor cannot find a parent and the network becomes disconnected.

Key Applications

An increasing number of applications such as environmental monitoring, habitat and seismic monitoring and surveillance, require the deployment of small, short-range and inexpensive sensor nodes in a field with the purpose of collecting data over long time periods. For example, biologists analyzing a forest are interested in the long-term behavior of the environment (forests, plant growth, animal movement, temperature). The purpose of the network is often to answer spatial queries such as "what are the moving patterns of the animals?" or "find the average temperature of the sensors in an area?". Real-time spatial data collection is also required in automated surveillance systems of open areas, roads and buildings for security purposes. These systems are useful in military and non-military settings, to detect and track the movement of people, report suspicious behavior and identify threats.

Future Directions

Several opportunities for future research can be identified. Spatial indexes such as SPIX have been designed to respond to spatial queries that are submitted from the base station to a sensor network. More sophisticated structures have to be developed so that they can be used to respond to spatial queries submitted from any of the nodes in the network. This will allow the use of such structures for surveillance applications such as object tracking though mobile users or sensors.

Cross References

► Retrieval Algorithms, Spatial

Recommended Reading

1. Beckman, N., Kriegel, H., Schneider, R., Seeger, B.: The R*tree: An efficient and robust access method for point and rectangles. ACM SIGMOD Record. **19**(2), 322–331 (1990)
2. Bonnet, P., Gehrke, J., Seshardi, P.: Toward sensor database systems. Paper presented at the 2nd International Conference on Mobile Data Management (MDM 2001), Hong Kong, China, January 2001, pp. 3–14
3. Demirbas, M., Ferhatosmanoglu, H.: Peer-to-peer spatial queries in Sensor Networks. Paper presented at the 3rd IEEE International Conference on Peer-to-Peer Computing (P2P'03), Linkoping, Sweden, September 2003, pp. 32–39
4. Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J., Silva, F.: Directed Diffusion for Wireless Sensor Networking. IEEE/ACM Transactions on Networking. **11**(1), 2–16 (2003)
5. Gutman, A.: RTree – A dynamic index structure for spatial searching. Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data (SIGMOD 1984), Boston, MA, 1984
6. Heinzelman, W.R., Chandrakassan, A.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. Paper presented at the 33rd Hawaii International Conference of System Sciences, Maui, Hawaii, January 2000
7. Hu., H., Xu, J., Wang, W.S., Zheng, B., Lee, D.L., Lee, W.: Proactive Caching for Spatial Queries in Mobile Environments. Proceedings of the 21st International Conference on Data Engineering (ICDE 2005), Tokyo, Japan, April 2005
8. Ko, Y.-B., Vaidya, N.H.: Location-aided routing (LAR) in mobile ad hoc networks. Wireless Networks. **6**(4), (2000)
9. Madden, S., Franklin, M.J., Hellerstein, J.M., Hong, W.: The Design of an Acquisitional Query Processor for Sensor Networks. Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data (SIGMOD), San Diego, CA, June 2003, pages 491–502
10. Sellis, T.K., Roussopoulos, N., Faloutsos, C.: The R+tree: A dynamic index for multi-dimensional objects. Proceedings of 13th International Conference on Very Large Data Bases (VLDB 1987), Brighton, England, Sept 1987
11. Soheili, A., Kalogeraki, V., Gunopulos, D.: Spatial Queries in Sensor Networks. Paper presented at the 13th International Symposium on Advances in Geographic Information Systems (GIS 2005), Bremen, Germany, November 2005
12. Tao, Y., Papadias, D., Spatial Queries in Dynamic Environments, ACM Trans. Database Syst. **28**(2), 101–139 (2003)

Data Compression

► Image Compression

Data Compression for Network GIS

HAIJUN ZHU, CHAOWEI PHIL YANG
Joint Center for Intelligent Spatial Computing (CISC), College of Science, George Mason University, Fairfax, VA, USA

Synonyms

Information Theory; Raster Data Compression; Non-Raster Data Compression

Definition

Data compression of Network GIS refers to the compression of geospatial data within a network GIS so that the volume of data transmitted across the network can be reduced. Typically, a properly chosen compression algorithm can reduce data size to 5～10% of the original for images [1,2], and 10～20% for vector [3] and textual data [4]. Such compression ratios result in significant performance improvement.

Data compression algorithms can be categorized into lossless and lossy. Bit streams generated by the lossless compression algorithm can be faithfully recovered to the original data. If loss of one single bit may cause serious and unpredictable consequences in the original data (for example, text and medical image compression), the lossless compression algorithm should be applied. If data consumers can tolerate distortion of the original data to a certain degree, lossy compression algorithms are usually better because they can achieve much higher compression ratios than lossless ones. Some commonly used lossless and lossy data compression algorithms are listed in Table 1.

Practical data compression applications do not have to be restricted to a single type. For example, the JPEG (Joint Photographic Expert Group) image compression [5] first uses DCT (Discrete Cosine Transform) to decompose images into transform coefficients. These transform coefficients are lossy quantized and the quantized coefficients are losslessly compressed with Huffman or arithmetic coding.

Web-based platforms pose new challenges for data compression algorithms because web users are pretty diversified in terms of number, objective, and performance tolerance. Within such a context, data compression algorithms should be robust and fast while consuming server resource as little as possible. Progressive transmission (PT) was proposed for such requirements [6,7,8]. A PT-enabled bitstream acts as a finite decimal number (e. g., 3.23897401), which, if decimated from the beginning to certain place (e. g., 3.2389), will result in a shorter bitstream that can be reconstructed to a low-precision version of original data. Only one version of PT-enabled bitstream needs to be stored and all lower precision bitstreams can be obtained therein.

PT is based on multiresolution data decomposition [6,7,8]. For raster data, many effective algorithms can be used to generate such decomposition. For non-raster data, it is quite hard to construct progressive bitstreams effectively because these data are not defined in a regular spatial grid and commonly used multi-resolution decomposition algorithms (e. g., wavelet) are difficult to apply [9,10,11]. Therefore, other methods (e. g., cartographical-principle based decimation, Fig. 1b and c may be adopted.

Data Compression for Network GIS, Table 1 Lossless and lossy data compression algorithms

Lossless	Lossy
Huffman Coding	Differential Pulse Coded Modulation (DPCM)
Arithmetic Coding	Transform Coding
Lempel-Ziv Coding (LZC)	Subband Coding
Burrows-Wheeler Transform (BWT)	Vector Quantization
…	…

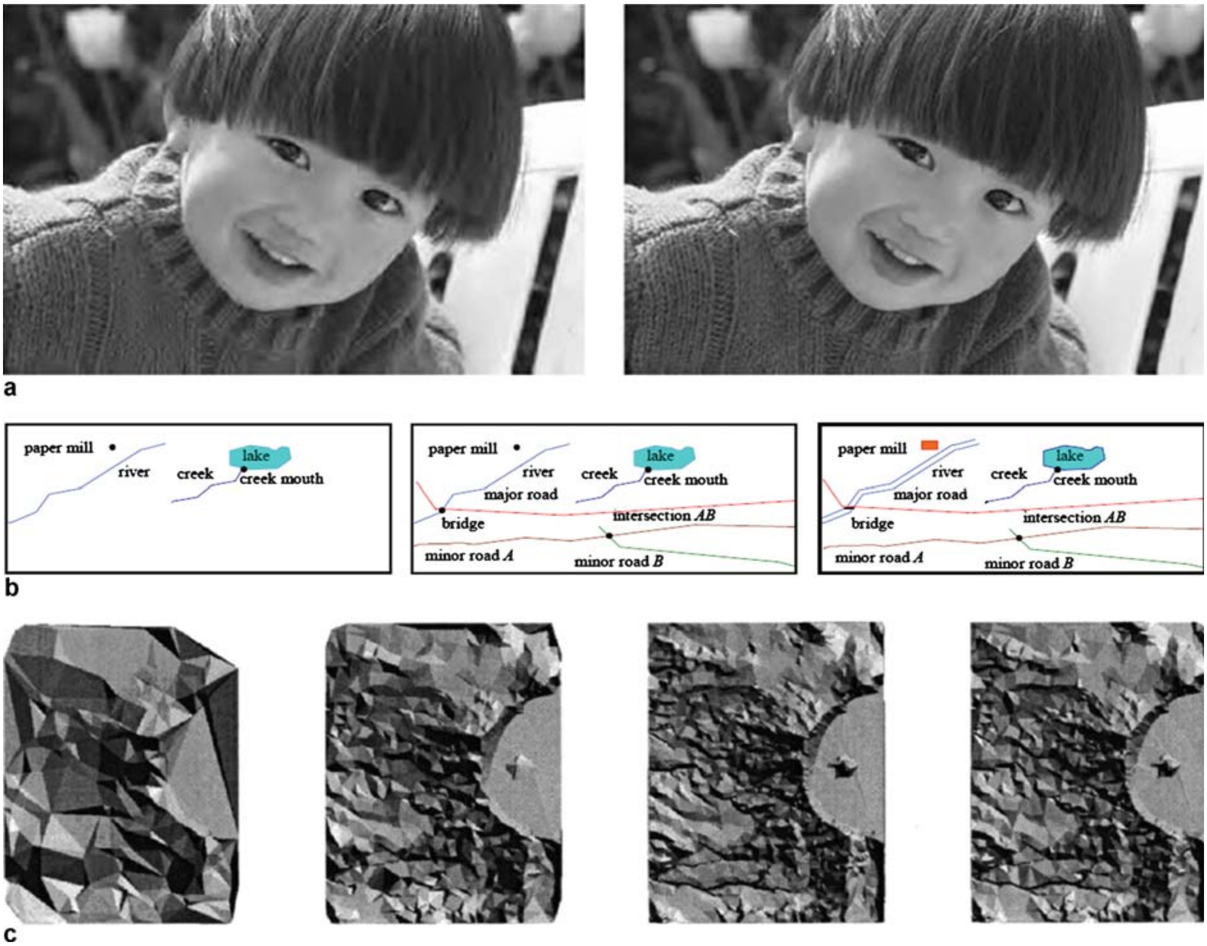
Historical Background

Data compression of network GIS is similar to other data compression algorithms on distributed computing platforms. Image compression algorithms such as JPEG had been applied since the first Web-based GIS emerged in 1993 [12]. However, the compression of vector data was introduced much later, such as the Douglas-Peuker algorithm [13] and the work done in 2001 by Bertolotto and Egenhofer [9].

Scientific Fundamentals

Data compression originates from information theory [14], which concentrates on the systematic research on problems arising when analog signals are converted to and from digital signals and digital signals are coded and transmitted via digital channels. One of the most significant theoretical results in information theory is the so-called source coding theorem [14], which asserts that there exists a compression ratio limit that can only be approached but never be exceeded by any compression algorithms. For the most practical signals, it is even very difficult to obtain compression algorithms whose performance is near this limit. However, compression ratio is by no means the unique principal in the development of the compression algorithm. Other important principals include fast compression speed, low resource consumption, simple implementation, error resilience, adaptability to different signals, etc. Further study regarding information theory and data compression can be found in texts [2,15] and journals (e. g., *IEEE Transactions on Information Theory*).

Progressive transmission algorithms are mostly based on wavelet decomposition, especially in digital images. In wavelet decomposition, signals are represented as a weighted sum of a group of wavelet bases. These bases are fast-decaying in both the spatial and frequency domain, which makes the analysis of local properties of signal effective. An example of image wavelet decomposition is



Data Compression for Network GIS, Figure 1 Progressive transmission of different types of data in networked GIS. **a** Image. **b** Cartographic principle based progressive vector. **c** TIN

illustrated in Fig. 2. Since wavelet decomposition is recursive, a progressive transmission algorithm can be immediately constructed by transmitting frequency bands successively from low to high. Other more efficient progressive transmission schemas may utilize the similarity between frequency bands [6,7] or optimally add more truncation points [8].

Key Applications

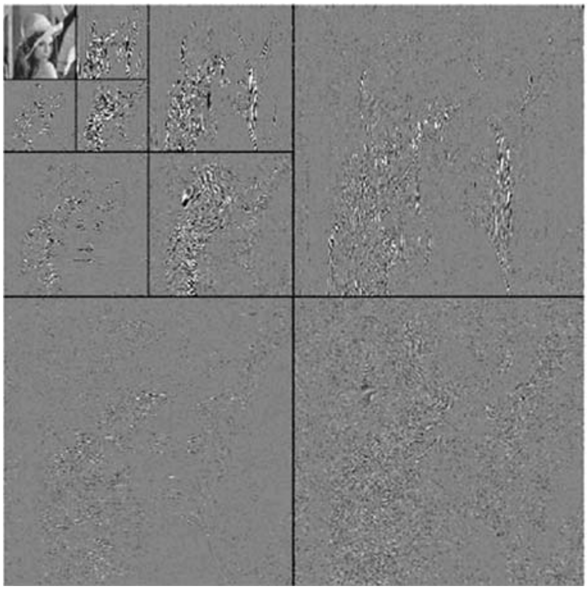
Raster Data Compression

Raster data compression algorithms are the same as algorithms for compression of other image data. However, geospatial images are usually of much higher resolution, multi-spectral and of a significant larger volume than natural images. To effectively compress raster data in networked GIS, emphasis must be put on the following aspects:

- Statistical properties of imagery in GIS may be quite different from other types of imagery,
- Correlation among different spectrums,
- Managing schemas [16] to deal with large volumes of geospatial raster data,
- Integration of other types of datasets (e. g., vector and 3-D data).

WebGIS:

- TerraServer [24] uses the so-called pyramid technique to assist the SQL Server to manage images. With this technique, a relatively large image is extracted into different levels of detail to construct a pyramid structure. The images are transmitted only when the data of interest are requested by the user.
- ArcGIS also uses the pyramid technique in handling big images and the pyramid is built on the fly every time when the image is accessed. However, this method is not suitable for managing images on WebGIS because



Data Compression for Network GIS, Figure 2 Wavelet decomposition of image

the response time will be too long. Yang, et al. [16] developed a method to manage a permanent pyramid so that performance can be improved.

- Google Earth [25] divides remote sensing images into many slices and organizes each slice into different resolutions using the progressive transmission method. Additionally, some Web2.0 techniques (e. g., AJAX) are incorporated so that user experience can be improved.

Non-raster Data Compression

Different methods can be utilized to compress non-raster data, such as 2-D and 3-D vector data (e. g., roads and borders), 3-D mesh models, and TIN.

For vector data, a survey of simplification algorithms can be found in [17]. Simplification aims at extracting a subset of original vector data according to predefined criteria. Resulting vector data is also compressed. Algorithms that derive binary coding for vector data [3,18] also exist. Compression algorithms for vector data are far less than those for raster data. Various research on progressive vector transmission algorithms concentrates more on topological and semantic aspects than pure binary coding [9,10,11]. However, due to the complexity of this problem, existing solutions are far from satisfactory.

For 3-D mesh models, usually the structure and attribute information are coded separately. Structure information records how vertices are connected and must be losslessly compressed. Attribute information records information for each single vertex and can be lossy compressed. Progressive mesh transmission algorithms [19] depend on how to decimate vertices one by one so that a given error criterion can be optimized.

Compression and progressive transmission of TIN is similar to 3-D mesh models [20].

GIS Interoperability

Interoperability gains popularity by sharing geospatial resources. However, the standardization of interoperable interfaces increase the volume of data that has to be transmitted. Therefore, the compression methods associated with interoperable encoding language are very important. For example, GML could be several times larger than the original data in binary format [21]. Possible solutions to such problems include:

1. A BLOB (Binary Large Object) object can be embedded in the textual XML document to store the binary compressed stream of geospatial data
2. A textual XML file can be compressed using common compression tools (e. g., zip and gzip) before transmitting
3. The BXML (Binary eXtensible Markup Language) proposed by CubeWerx Inc. [22] and the OGC (Open GIS Consortium) also provides promising results [23].

Future Directions

Future research needed in this area includes 1) principals and algorithms for optimally choosing proper compression schemas and parameters when compressing raster data, 2) semantically and topologically well-designed progressive transmission algorithms for non-raster data, and 3) incorporating proper compression algorithms for both raster and non-raster data into different Web infrastructures.

Cross References

► Network GIS Performance

Recommended Reading

1. Egger, O., Fleury, P., Ebrahimi, T., Kunt, M.: High-performance compression of visual information-a tutorial review-part I: still pictures. In: Proc. IEEE, vol. 87, pp. 976–1011. June 1999
2. Jayant, N.S., Noll, P.: Digital coding of waveforms. Prentice-Hall, Englewood Cliffs, NJ (1984)
3. Shekhar, S., Huang, Y., Djugash, J., Zhou, C.: Vector map compression: a clustering approach. In: Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems, McLean, Virginia, 8–9 Nov 2002
4. Bell, T., Witten, I.H., Cleary, J.G.: Modeling for text compression. ACM Computing Surveys. **21**, 557–591 (1989)

5. Information technology – digital compression and coding of continuous-tone still images – Part 1: Requirements and guidelines. ISO/IEC International Standard 10918-1, ITU-T Rec. T.81 (1993)
6. Shapiro, J.M.: Embedded image coding using zerotrees of wavelet coefficients. IEEE Trans. Signal Processing **41**, 3445–3462 (1993)
7. Said, A., Pearlman, W.A.: A new, fast, and efficient image codec based on set partitioning in hierarchical trees. IEEE Trans. Circuits and Systems for Video Technology **6**, 243–250 (1996)
8. Taubman, D.: High performance scalable image compression with EBCOT. IEEE Trans. Image Processing **9**, 1158–1170 (2000)
9. Bertolotto, M., Egenhofer, M.J.: Progressive transmission of vector map data over the World Wide Web. GeoInformatica **5**(4), 345–373 (2001)
10. Buttenfield, B.P.: Transmitting vector geospatial data across the Internet, Proceedings of the Second International Conference on Geographic Information Science, 25–28 September 2002, Boulder, Colorado, USA, Lecture notes in Computer Science 2478. Springer, Berlin (2002) pp. 51–64
11. Bertolotto, M., Egenhofer, M.J.: Progressive vector transmission. In: Proceedings of the 7th International Symposium on Advances in GIS, Kansas City, Nov 1999
12. Plewe, B.: GIS Online. Information retrieval, mapping and the Internet. OnWord Press, Albany, New York (1997)
13. Douglas, D., Peucker, T.: An algorithm for the reduction of the number of points required to represent a digitized line or its character. The Canadian Cartographer **10**(2), 112–122 (1973)
14. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**, 379–423 623–656 (1948)
15. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley, Chichester, U.K. (1991)
16. Yang, C., Wong, D., Yang, R., Kafatos, M., Li, Q.: Performance improving techniques in WebGIS. International Journal of Geographic Information Sciences **19**(3), 319–342 (2005)
17. Heckbert, P.S., Garland, M.: Survey of polygonal surface simplification algorithms. Carnegie Mellon University, Pittsburg, Pennsylvania (1997)
18. Lu, C.C., Dunham, J.G.: Highly efficient coding schemes for contour lines based on chain code representations. IEEE Transaction on Communication **39**(10), 1511–1514 (1991)
19. Hoppe, H.: Progressive meshes. In: SIGGRAPH'96 Proceedings, pp. 99–108. New Orleans, Louisiana, USA 4–9 Aug 1996
20. Park, D., Cho, H., Kim, Y.: A TIN compression method using Delaunay triangulation. Int. J. Geographical Information Science **15**, 255–269 (2001)
21. OGC 03-002r9 (2006) Binary Extensible Markup Language (BXML) Encoding Specification, version 0.0.8, Craig Bruce, http://www.opengeospatial.org/standards/bp. Accessed 20 Aug 2006
22. CubeWerx Inc. http://www.cubewrex.com/. Accessed 20 Aug 2006
23. OGC 05-50 (2005) GML Performance Investigation by CubeWerx, version 1.0.0, Craig Bruce, http://www.opengeospatial.org/standards/gml. Accessed 20 Aug 2006
24. Barclay T, Gray J,Slutz D, (1999) Microsoft TerraServer: A Spatial Data Warehouse, Microsoft Technical Report, http://research.microsoft.com/users/Tbarclay/MicrosoftTerraServer_TechnicalReport.doc. Accessed 20 Aug 2006
25. Google Earth. http://earth.google.com. Accessed 20 Dec 2006

Data Cube

► OLAP, Spatial

Data Grid

► Grid

Data Infrastructure, Spatial

FREDERICO FONSECA
College of Information Sciences and Technology,
The Pennsylvania State University,
University Park, PA, USA

Synonyms

Map distribution; Digital divide

Definition

Spatial data infrastructure (SDI) is the well connected and functional assemblage of the required technology, policies, and people to enable the sharing and use of geographic information. It should include all levels of organizations and individuals such as government agencies, industry, nonprofit organizations, the academic community, and individuals.

Historical Background

The creation of the concept of SDI is the result of the increasing availability of geospatial data. This increase is due mostly to the spread of the technology, which leads to lower costs. Growing concerns with the environment also lead many governments to start distributing data and creating policies that allow for a broader access to the data. Along with the availability of data soon came an awareness that the technology was difficult to handle for most end users. Geospatial data by itself is not enough to enable users to effectively reason and make decisions on environmental issues. Therefore, SDI came to be seen in two different ways.

First, it came to be seen as an automated map distribution system. In this case, the implementation of a SDI focuses on map production and distribution of existing sources on an "as-is" basis. A second view is to see SDI as an enabler for understanding space. In this case, SDI does not only deliver maps, but disseminates spatial data with associated quality control, metadata information, and semantic descriptions. The SDI user is someone who is able to combine spatial data from different sources to produce new information for a study area. This second vision is the one where SDI can play an important role in creating an effective use of geospatial information at the different levels. While it is important in the long term to provide users with efficient means to feed their own creations, such as digital maps or analysis results, back into an overall SDI cataloging, archiving, search and retrieval system, the core of an SDI resides in its source data

Scientific Fundamentals

Digital Divide

The digital divide is defined as the gap between those with regular, effective access to digital technologies and those without. Spatial data, without an adequate SDI, will only make the gap wider. The complexity of handling geospatial information and reasoning about it is compounded by the steep prices of hardware and software necessary to store and process the data.

Education

Geographic information systems (GIS) is considered a disruptive technology. Such technologies are new technologies that require important organizational changes. They usually need specialists and managers whose knowledge is very different from that of those who used the technology it displaces. This is clearly the case in GIS, where manual map makers are replaced by geographical database specialists. Disruptive technologies, such as GIS, are usually actively promoted by software developers and service vendors. Such "push-oriented" actions are not matched by the ability of users to adapt to the technological change. It is also necessary to recognize the importance of dealing with spatial information as a fundamental part of information infrastructure, and not as a collection of digital maps.

Open Access to Data

As the user base of GIS/SDI expands, new users are likely to have a more application-oriented profile. Increasing demand for high-quality spatial data is likely to force all players to clearly establish their data policies. In the long run, SDI may be facing a dilemma between having either good data commercially available but out of reach of a large number of users, or free data of low quality.

Open Access to Software

One of the main concerns in the establishment of SDI is the issue of avoiding the "lock-in" effect in the choice of technology. This effect is well known in the software industry, since the customer may become dependent on proprietary data formats or interfaces, and high switching costs might prevent change to another product. Substantial barriers to entry are created, resulting in effective monopolies. Globally, the GIS software market has a tendency towards an oligopoly in which very few companies have a large market share. SDI could benefit from the emergence of open-source GIS to produce solutions that match user needs and avoid proprietary technology. Open source GIS software such as PostGIS, MapServer and TerraLib can provide an effective technological base to develop SDI that are independent of proprietary technology. GIS open-source software tools allow researchers and solution providers to access a wider range of tools than is currently offered by the commercial companies.

Key Applications

SDI is most needed as a support for decision making. For democratic access to geospatial information, it is necessary that all the players in the decision have full access and understanding of the information on discussion. For example, planning a new hydroelectric power plant requires an assessment of its potential impacts on communities and the environment. This leads to a need for building different scenarios with quality spatial data and adequate spatial analysis techniques. Static map products are unsuitable for such analyses. Thus, SDI will only have an impact if all players involved in the decision process are knowledgeable about GIS technology.

Future Directions

For SDI, low-cost or open-source software is crucial. GIS software development is changing. Coupled with advances in database management systems, rapid application development environments enable building "vertically integrated" solutions tailored to the users' needs. Therefore, an important challenge for the GIS/SDI community is finding ways of taking advantage of the new generation of spatially enabled database systems to build "faster, cheaper, smaller" GIS/SDI technology. In order to make the applications work, high quality data is also necessary. In the long run, it is necessary not only to implement SDI, but also to make it sustainable. In this case, sustainability means that the SDI will work, in practice, over time, in a local setting. The SDI has to be adapted to the different contexts, learning with locals and adopting practices that will persist over time. It is necessary that governments put in place policies that enforce the availability of data and software. Open access to both of them have to be enforced by policies that make sure that the digital divide in spatial data is avoided.

Recommended Reading

1. Câmara, G., Fonseca, F., Monteiro, A.M., Onsrud, H.: Networks of innovation and the establishment of a spatial data infrastructure in Brazil. Inf. Technol. Dev. **12**, 255–272 (2006)

2. Onsrud, H., Poore, B., Rugg, R., Taupier, R., Wiggins, L.: The future of the spatial information infrastructure. In: McMaster, R.B., Usery, E.L. (eds.) A Research Agenda for Geographic Information Science, pp225–255. CRC, Boca Raton (2004)
3. Aanestad, M., Monteiro, E., Nielsen, P.: Information infrastructures and public goods: Analytical and practical implications for SDI. Inf. Technol. Dev. **13**, 7–25 (2007)
4. Davis, C.A., Fonseca, F.: Considerations from the development of a local spatial data infrastructure. Inf. Technol. Dev. **12**, 273–290 (2006)
5. Georgiadou, Y., Bernard, L., Sahay, S.: Implementation of spatial data infrastructures in transitional economies. Inf. Technol. Dev. **12**, 247–253 (2006)
6. Lance, K., Bassolé, A.: SDI and national information and communication infrastructure (NICI) integration in Africa. Inf. Technol. Dev. **12**, 333–338 (2006)
7. Satish, K.P.: Technological frames of stakeholders shaping the SDI implementation: A case study from India. Inf. Technol. Dev. **12**, 311–331 (2006)
8. Sorensen, M., Sayegh, F.: The initiation, growth, sustainment of SDI in the Middle East—notes from the trenches. Inf. Technol. Dev. **13**, 95–100 (2007)

Data Integration

► Conflation of Features
► Ontology-Based Geospatial Data Integration

Data Modeling

► Modeling and Multiple Perceptions

Data Models

► Application Schema
► Data Models in Commercial GIS Systems

Data Models in Commercial GIS Systems

ERIK HOEL
ESRI, Redlands, CA, USA

Synonyms

Data models; Data representations; Vector Models; Raster Models

Definition

Geographic data models are used to represent real world objects (e. g., buildings, roads, land parcels, rainfall, soil types, hills and valleys, etc.) within a geographic information system. These data models are used by the GIS to perform interactive queries, execute analyses, and produce cartographic maps. Many different data types may be used to model this data. Commercial GIS systems are intended to address diverse user requirements across a broad spectrum of application domains. Some users of these systems are focused on traditional two-dimensional vector representations of spatial data (e. g., modeling topologically integrated cadastres, road networks, or hydrologic networks), while other users are concerned with raster data obtained from satellite imagery and other aerial image sources. In addition, the advent of Light Detection and Ranging has provided large sources of z-enabled data facilitating the very accurate modeling of 2.5D surfaces. In many domains, vector, raster, and surfaces are used in conjunction with one and other in order to support sophisticated visualization and analysis. As such, commercial GIS systems must support a large variety of different data models in order to meet these widely differing requirements.

Historical Background

Data models are a core aspect of all GIS systems starting with the earliest systems (CGIS – Canadian Geographic Information Systems, 1964). Some of the earliest nontrivial (e. g., data models where the geometries representing the real world features are related in some explicit manner) include topological models such as GBF-DIME (US Census Bureau,1967), POLYVRT (Harvard Laboratory for Computer Graphics, 1973), and TIGER (US Census Bureau, 1986). Others, such as the Minnesota Land Management Information System (MLMIS) in the 1960s represented geographical data with rasters. These early data models have directly contributed to the development of today's sophisticated GIS data models.

Scientific Fundamentals

In the context of GIS systems, a data model is a mathematical construct for representing geographic objects or surfaces as data. For example, the vector data model represents geography as collections of points, lines, and polygons; the raster data model represents geography as cell matrixes that store numeric values. Surface data models represent surface geography in either raster (sets of regularly spaced cells) or vector (sets of irregularly distributed mass points) formats.

Vector Models

A coordinate-based data model that represents geographic features as points, lines, and polygons. Each point feature is represented as a single coordinate pair, while line and polygon features are represented as ordered lists of vertices. Attributes are associated with each vector feature, as opposed to a raster data model, which associates attributes with grid cells. Vector models are useful for storing data that has discrete boundaries, such as country borders, land parcels, and streets.

Vector models can be categorized into several different subtypes:

- Spaghetti models
- Network models
- Topological models

Spaghetti Models

Spaghetti models (sometimes termed simple data models) are the simplest of the vector-based models where the geometric representations of spatial features do not have any explicit relationship (e. g., topological or network) to any other spatial feature. The geometries may be points, lines, or polygons. There are no constraints with respect to how geometries may positioned – e. g., two lines may intersect without a point being positioned at the location of intersection, or two or more polygons may intersect without restriction.

Spaghetti models may offer several advantages over other data models. These advantages include simplicity of the model, ease of editing, and drawing performance. The disadvantages of the spaghetti model include the possible redundant storage of data and the computational expense in determining topological or network relationships between features. In addition, spaghetti models cannot be used to effectively represent surface data.

Network Models

Networks are used to model the transportation of people and resources such as water, electricity, gas, and telecommunications. Networks are a one-dimensional collection of topologically interconnected point and line features (commonly termed junctions and edges respectively), where the edges connect to junctions. Network commonly facilitate the modeling of constrained flow along edges (such as streets and river reaches) and through junctions (such as intersections and confluences).

Within the network model domain, there are two fundamental subtypes of networks: those where the flow is undirected, and those where the flow is directed. As an example of each type, transportation network are generally considered to be undirected, while utility or natural resource networks (e. g., river networks) are modeled using directed networks.

Directed Network Models

Directed network models are typically used to model directed flow systems. These are systems where a resource moves in one direction through the edges in the network. Common applications for directed networks include the modeling of hydrologic (river) networks as well as utility networks.

Network elements (edges and junctions) within a directed network are commonly associated with collections of attributes. These attributes on the network elements may be used for modeling flow direction, classifications (e. g., pipe type), restrictions (e. g., maximum flow), and impedances.

Directed Network Models – Hydrologic Networks

Rainfall on the landscape accumulates from rivulets to streams, rivers, and finally, an ocean. The shape of the surface directs water to a stream network. Gravity drives river flow from higher elevations to sea level. A hydrologic network usually models a river as a connected set of directed stream reaches (edges) and their confluences (junctions). When a stream drains into a lake, hydrologic models continue the flow along an arbitrary line midway between shores until an outlet is reached.

Special large-scale hydrologic project models may include 3D analysis of flow lines through a channel volume, but simplifying a river to a one-dimension line network is suitable for most applications. In flat terrain, river flow becomes more complicated – a large river near an ocean often forms a delta with a complex braided network and tidal effects can reverse flow near the shore.

Some common tasks on hydrologic networks include:

- Deriving catchments on a surface model for each stream reach
- Accumulating rainfall on catchments, transfer flow to reach
- Using gauge valves, predict flood surge along a river
- Design a system of channels and holding ponds for high water
- Managing diversion of water for agriculture or city water works

Directed Network Models – Utility Networks

Utility networks (modeled on top of directed networks) are the built environment that supplies energy, water, and communications and removes effluent and storm water. Water utilities are gravity driven or pressurized, depending on terrain. Flow in a gas utility is driven by pressure in pipes. Electric power flows from high voltage potential to low. Pulses of light carry communications in a fiber optic network.

Utility networks have a nominal flow condition, with a few sources delivering a resource to many points of consumption. Some utility networks tolerate loops, such as a water network. For other utilities, a loop is a fault condition, such as an electrical short circuit. All utility networks contain dynamic devices such as valves and switches that can interrupt or redirect flow in the event of an outage or system maintenance.

Some utilities such as telecommunications and electrical networks have multiple circuits on a common carrier (edge), such as electric lines with three phases of power or twisted-pair lines in telephony.

Some utility network tasks are:

- Establishing the direction of a commodity flow
- Finding what is upstream of a point
- Closing switches or valves to redirect flow
- Identifying isolated parts of the network
- Finding facilities that serve a set of customers

Undirected Network Models

Undirected networks, the second type of network model, are most commonly used to model transportation. Transportation involves the movement of people and the shipment of goods from one location to another. Transportation networks are the ubiquitous network – people commonly spend a fraction of every day traversing this network. Transportation networks have two-way flow, except for situations such as one-way streets, divided highways, and transition ramps.

As with directed networks, network elements (edges and junctions) are commonly associated with collections of attributes. These attributes on the network elements may be used for modeling classifications (e. g., road type), restrictions (e. g., pedestrian traffic not allowed), and impedances (e. g., drive times). Differing from common directed networks, transportation networks also need the ability to represent turn restrictions and turn impedances (e. g., the cost of turning across oncoming traffic at an intersection).

Transportation networks often form a multi-level network—while most roads are at surface level, bridges, tunnels, and highway interchanges cross each other in elevation; a simple overpass has two levels and a highway interchange typically has four.

When moving through a transportation network traveling, people optimize the process by hopping from one mode of transport to another; e. g., switching between walking, driving, riding a bus or train, and flying. There are also natural hierarchies in transportation network. Trips of any distance usually begin by driving to the closest freeway on-ramp and proceeding to the off-ramp closest to the destination.

Common transportation network related tasks are:

- Calculating the quickest path between two locations
- Determining a trade area based upon travel time
- Dispatching an ambulance to an accident
- Finding the best route and sequence to visit a set of customers
- Efficiently routing a garbage truck or snow plow
- Forecast demand for transportation

Topology Models

Topology has historically been viewed as a spatial data structure used primarily to ensure that the associated data forms a consistent and clean topological fabric. Topology is used most fundamentally to ensure data quality (e. g., no gaps or overlaps between polygons representing land parcels) and allow a GIS to more realistically represent geographic features. Topology allows you to control the geometric relationships between features and maintain their geometric integrity.

The common representation of a topology is as a collection of topological primitives – i. e., nodes, arcs, and faces, with explicit relationships between the primitives themselves. For example, an arc would have a relationship to the face on the left, and the face on the right. With advances in GIS development, an alternative view of topology has evolved. Topology can be modeled as a collection of rules and relationships that, coupled with a set of editing tools and techniques, enables a GIS to more accurately model geometric relationships found in the world.

Topology, implemented as feature behavior and user specified rules, allows a more flexible set of geometric relationships to be modeled than topology implemented as a data structure. For example, older data structure based topology models enforce a fixed collection of rules that define topological integrity within a collection of data. The alternative approach (feature behavior and rules) allows topological relationships to exist between more discrete types of features within a feature dataset. In this alternative view, topology may still be employed to ensure that the data forms a clean and consistent topological fabric, but also more broadly, it is used to ensure that the features obey the key geometric rules defined for their role in the database.

Raster Models

A Raster Model defines space as an array of equally sized cells arranged in rows and columns, and comprised of single or multiple bands. Each cell contains an attribute value and location coordinates. Unlike a vector model which stores coordinates explicitly, raster coordinates are contained in the ordering of the matrix. Groups of cells that share the same value represent the same type of geographic feature. Raster models are useful for storing data that varies continuously, as in an aerial photograph, a satellite image, a surface of chemical concentrations, or an elevation surface. Rasters can also be used to represent an imaged map, a 2.5D surface, or photographs of objects referenced to features.

With the raster data model, spatial data is not continuous but divided into discrete units. This makes raster data particularly suitable for certain types of spatial operations, such as overlays or area calculations. Unlike vector data, however, there are no implicit topological relationships.

A band within a raster is a layer that represents data values for a specific range in the electromagnetic spectrum (such as ultraviolet, blue, green, red, and infrared), or radar, or other values derived by manipulating the original image bands. A Raster Model can contain more than one band. For example, satellite imagery commonly has multiple bands representing different wavelengths of energy from along the electromagnetic spectrum.

Rasters are single images that are stored in the GIS. These images may be as simple as a single image imported from a file on disk to a large image that has been created by mosaicing or appending multiple images together into a single, large, and seamless image. MrSIDs, GRIDs, TIFFs, and ERDAS Imagine files are all examples of rasters.

Raster Catalogs

A Raster Catalog (or an image catalog) is an extension to the raster model where a collection of rasters are defined in a table of any format, in which the records define the individual rasters that are included in the catalog. Raster catalogs can be used to display adjacent or overlapping rasters without having to mosaic them together into one large file. Raster catalog are also sometimes called image catalogs.

Each raster in a raster catalog maintains its own properties. For example, one raster might have a different color map than another raster, or one might have a different number of bands than another. Raster Catalogs can accommodate a different color map for each raster.

A raster inside a raster catalog behaves in the same way as a stand-alone raster dataset. Therefore, you can mosaic raster data into a raster that resides in a raster catalog.

A raster catalog model should be used when:

- Overlapping areas of individual inputs are important
- Metadata of individual inputs is important
- Query on attributes/metadata (i. e., percentage cloud cover)
- Simply want to keep/store individual images

Surface Models

Surface Models (or digital elevation models – DEMs) are used to represent the surface topography of the earth. Surface models are commonly used for creating relief maps, rendering 3D visualizations, modeling water flow, rectification of aerial photography, and terrain analyses in geomorphology.

Surface models are commonly built from remote sensing data (e. g., synthetic aperture radar [SAR] and light detection and ranging [LIDAR]) or from traditional survey methods. Surface models come in two primary forms depending upon the type of source data being used for their construction. Raster-based models are surfaces made from regularly spaced elevation measurements. The second type is vector-based (usually termed Triangular Irregular Network, or TIN). These vector-based are surfaces made from irregularly distributed measurement points (termed mass points).

Surface Models – Raster-based

With Raster-based surface models (sometimes termed GRIDs), the source elevation data forms a regularly spaced grid of cells. The size of the cells is fixed within the model. Common cell sizes vary between 25 and 250 meters. In a grid cell, the elevation of the corresponding geographic area is assumed constant. The USGS DEM and the DTED (Digital Terrain Elevation Data) are notable raster-based surface model standards.

Surface Models – Vector-based

TINs (Triangulated Irregular Networks) are a vector data structure that partitions geographic space into contiguous, non-overlapping triangles. The vertices of each triangle are sample data points with x-, y-, and z-values (used to represent elevations). These sample points are connected by lines to form Delaunay triangles. A TIN is a complete planar graph that maintains topological relationships between its constituent elements: nodes, edges, and triangles. Point, line, and polygon features can be incorporated into a TIN. The vertices are used as nodes which are connected by edges that form triangles. Edges connect nodes that are close to one another.

The partitioning of continuous space into triangular facets facilitates surface modeling because a very close approximation of a surface can be made by fitting triangles to planar, or near planar, patches on the surface. Input vector data is incorporated directly in the model and any resulting query or analysis will honor them exactly. Since the triangulation is based on proximity, interpolation neighborhoods are always comprised of the closest input data/samples. Proximity based connectivity is useful for other analysis as well. For example, Thiessen polygons, also known as Voronoi diagrams, are constructed from TINs.

Key Applications

Geographic data models are used in GIS systems to represent and model real-world entities. Oftentimes, collections of base data models (e. g., networks and topologies) are combined into larger, more complex data models, with the base models representing thematic layers within the larger model. The types of applications are extremely diverse; a small collection of examples and key applications include:

- Simple vector models can be used to model biodiversity conservation models. More specifically, model the observed, predicted, and potential habitats for collections of threatened species in a study area.
- Hydrographic (water resource) data models can be assembled with river systems being modeled with directed networks, drainage areas (for estimating water flow into rivers) being modeled with topology models, surface terrain (for deriving rivers and drainage areas) using either raster or vector-based surface models.
- Modeling electric utilities with directed networks; this allows utilities to plan and monitor their outside plant equipment as well as perform analyses such as capacity planning as well as outage management (e. g., how to reroute power distribution during thunderstorms when devices are destroyed by lightning strikes).
- Using surface models to perform line-of-sight calculations for cell phone signal attenuation and coverage simulations (i. e., where should new towers be placed in order to maximize improvements in coverage).
- Employing simple vector data models to track crime or fire incidents, support command and control decision making, as well as model demographics, crime densities, and other hazard locations in public safety applications.
- Use a combination of simple, topological, surface, and raster models for forestry management. The models facilitate operation decision making (e. g., managing production costs), investment decision making (e. g., how to best invest in growing stock), and stewardship decision making (e. g., consideration of the ecosystem to allow the forest to grow).
- Using multiple data model types in order to create rich models that support homeland security activities. This include representing incidents (criminal activity, fires, hazardous material, air, rail, and vehicle), natural events (geologic or hydro-meteorological), operations (emergency medical, law enforcement, or sensor), and infrastructure (agriculture and fool, banking and insurance, commercial, educational facilities, energy facilities, public venues, and transportation).

Future Directions

Temporal Data

Data that specifically refers to times or dates. Temporal data may refer to discrete events, such as lightning strikes; moving objects, such as trains; or repeated observations, such as counts from traffic sensors. Temporal data is often bitemporal – meaning both valid-time (real world time) as well as transaction time (database time) need to be represented. It is not apparent that new data models need to be developed, but rather existing models (e. g., simple vector, network, topology, surface, or raster) need to be augmented to support more effective temporal indexing and analysis.

D

Cross References

► Oracle Spatial, Raster Data
► Spatial Data Transfer Standard (SDTS)
► Voronoi Diagram

Recommended Reading

1. Bernhardsen, T.: Geographic information systems: an introduction, 3rd ed. John Wiley & Sons, New York (2002)
2. Chrisman, N.: Charting the unknown: how computer mapping at Harvard became GIS. ESRI Press, Redlands, CA (2006)
3. Cooke, D., Maxfield, W.: The development of a geographic base file and its uses for mapping. In: 5th Annual Conference of the Urban and Regional Information System Association (URISA), pp. 207–218 (1967)
4. DeMers, M.: Fundamentals of geographic information systems, 3rd ed. John Wiley & Sons, New York (2005)
5. Hoel, E., Menon, S., Morehouse, S.: Building a robust relational implementation of topology. In: Proc. of Symposium on Advances in Spatial and Temporal Databases (SSTD), pp. 508–524 (2003)
6. Laurini, R., Thompson, D.: Fundamentals of spatial information systems. Academic Press, London (1992)
7. Longley, P., Goodchild, M., Maguire, D., Rhind, D.: Geographic information systems and science, 2nd ed. John Wiley & Sons, Chichester, England (2005)
8. Peucker, T., Chrisman, N.: Cartographic data structures. The American Cartographer, **2**(1), 55–69 (1975)
9. Rigaux, P., Scholl, M., Voisard, A.: Spatial databases with application to GIS. Morgan Kaufmann, San Francisco (2002)
10. Samet, H.: Foundations of multidimensional and metric data structures. Morgan Kaufmann, San Francisco (2006)
11. Schneider, M.: Spatial data types for database systems: finite resolution geometry for geographic information systems. Lecture Notes in Computer Science, vol. 1288, Springer Verlag, Berlin (1997)
12. Shekhar, S., Chawla, S.: Spatial databases: a tour. Prentice Hall, Upper Saddle River, NJ (2003)
13. Thill, J.-C.: Geographic information systems in transportation research. Elsevier Science, Oxford, England (2000)
14. Tomlinson, R.: Thinking about GIS: geographic information system planning for managers. ESRI Press, Redlands, CA (2003)
15. Zeiler, M.: Modeling our world: the ESRI guide to geodatabase design. ESRI Press, Redlands, CA (1999)

Data Quality

► Spatial Data Transfer Standard (SDTS)

Data Representations

► Data Models in Commercial GIS Systems

Data Schema

► Application Schema

Data Structure

Marci Sperber
Department of Computer Science and Engineering,
University of Minnesota, Minneapolis, MN, USA

Synonyms

Algorithm

Definition

A data structure is information that is organized in a certain way in memory in order to access it more efficiently. The data structure makes it easier to access and modify data.

Main Text

There are many types of data structures. Some examples are stacks, lists, arrays, hash tables, queues, and trees. There is not a data structure that is efficient for every purpose, so there are many different types to use for many different problems or purposes. A data structure should be chosen so that it can perform many types of operations while using little memory and execution time. An example of a good data structure fit would be using a tree-type data structure for use with a database. Of course there may be many data structures that can be used for a specific problem. The choice in these cases is mostly made by preference of the programmer or designer.

Cross References

► Indexing, Hilbert R-tree, Spatial Indexing, Multimedia Indexing
► Quadtree and Octree

Data Types for Moving Objects

► Spatio-temporal Data Types

Data Types for Uncertain, Indeterminate, or Imprecise Spatial Objects

► Vague Spatial Data Types

Data Warehouses and GIS

James B. Pick
School of Business, University of Redlands,
Redlands, CA, USA

Synonyms

Spatial data warehouses; Spatially-enabled data warehouses

Definition

The *data warehouse* is an alternative form of data storage from the conventional relational database. It is oriented towards a view of data that is subject-oriented, rather than application-oriented. It receives data from one or multiple relational databases, stores large or massive amounts of data, and emphasizes permanent storage of data received over periods of time. Data warehouses can be spatially enabled in several ways. The data in the warehouse can have spatial attributes, supporting mapping. Mapping functions are built into some data warehouse packages. Online analytical processing (OLAP) "slicing and dicing" and what-if functions are performed on the data in the warehouse, and may include spatial characteristics. Furthermore, the data warehouse can be linked to geographical information systems (GIS), data mining and other software packages for more spatial and numerical analysis. Data warehouses and GIS used conjointly emphasize the advantages of each, namely the large size, time variance, and easy arrangement of data in the warehouse, along with the spatial visualization and analysis capabilities of GIS.

Historical Background

Although databases and decision support systems existed in the 1960s and the relational database appeared in the 1970s, it was not until the 1980s that data warehouses began to appear for use [1]. By 1990, the concepts of data warehouse had developed enough that the first major data warehouse textbook appeared [2]. The analytical methods were a collection of methods drawn from statistics, neural networks, and other fields. The theory of the processing steps for data warehousing, OLAP, was formulated in 1995 by Codd [3]. The growth in the markets for data warehousing was driven by the expanding data storage and its analytical uses in organizations. During the past fifteen years, database companies such as Oracle and Sybase produced data warehousing products as well as computer vendors Microsoft and IBM, and enterprise resource planning (ERP) vendor SAP [1].

Early geographic information systems were associated with databases, but not until much later with data warehouses. In the past five years, some data warehouse products such as Oracle [4,5] became GIS-enabled. ERP products have been linked to leading GIS products. More common than tight integration of data warehouses and GIS is loose connections through data flows between data warehouses and GIS software.

Scientific Fundamentals

There are a number of scientific principles of data warehouses that are basic and also are related to GIS.

A data warehouse differs from a relational database in the following key characteristics: data warehouses are subject-oriented, time-variant, non-volatile, integrated, and oriented towards users who are decision-makers [1,6]. *Subject-oriented* means that the user accesses information in the data warehouse through common business subjects, such as part, customer, competitor, order, and factory. This contrasts with relational databases that often show the user many detailed attributes to access, but ones not necessarily of high user importance. The traditional operational database focuses on the functional areas of the business, such as sales, finance, and manufacturing.

Another data warehouse feature is that it retains older data, which makes possible analysis of change tendencies over time. An operational database regularly purges older data for deletion or archiving. Since the philosophy is to store data over long periods, the size of storage can be potentially huge, truly a "warehouse."

The warehouse data are *non-volatile*: after data are stored, they are fixed over time. This lends stability to the data in a data warehouse [2]. The data warehouse concept also favors the formation of summarized data, which are useful in decision-making and also become non-volatile. *Granularity* distinguishes the individual data items from the summaries: the most granular are raw data, while less granular are summarized data [1]. An example of summarized data is a summary of account totals for March, 2007, for a business department.

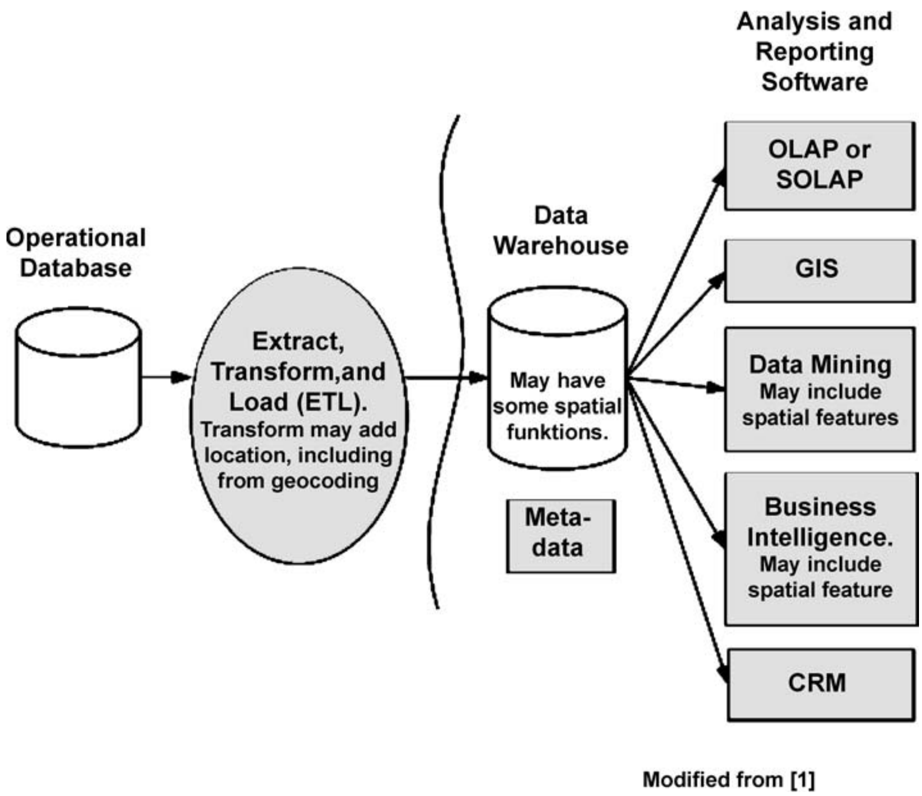
The data warehouse is *time-aggregated* also. For the data warehouse, data are extracted from multiple operational databases, made consistent, transformed, scrutinized for quality, and then appended to the data warehouse, with no further updating allowed of those data, i. e. they are non-volatile. At the next desired time point, data are again appended to the data warehouse and become non-volatile. Thus the data warehouse accumulates a time series of data by extracting them at multiple time points. Hence the data are "time aggregated", and remain available over a time period.

Data are extracted for the data warehouse from many sources, some of which are from legacy systems [7], combined together, and written to the data warehouse. Thus the data warehouse transforms the diverse data sources into an integrated data set for permanent, long-term storage. This process of transformation is referred to as *integration* [1]. The resultant data, which are of high quality, diverse in sources, and extending over long periods, are particularly suitable to analytical decision-makers, rather than operational transaction-based users.

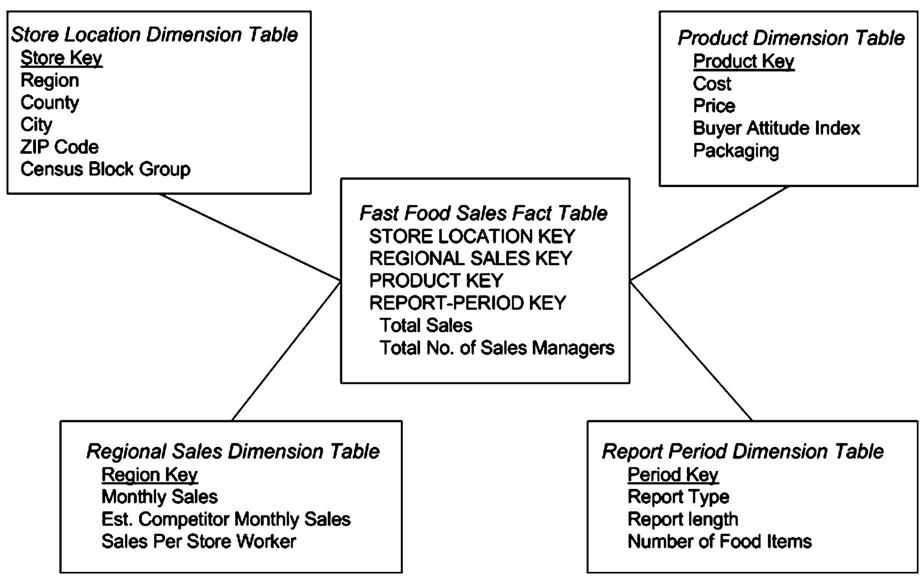
In a large organization, data are gathered and transformed from a wide collection of operational databases; data are checked for accuracy; and errors corrected. The whole process of input, extraction, error-checking, integration, and storing these data is known as "ETL" (extraction, transformation, and load). As shown in Fig. 1, after data enter the warehouse from the operational databases, they can be accessed by a variety of analytical, statistical, data mining, and GIS software [1,6]. Also shown are metadata, which keep track of the detailed descriptions of the records and entities in the data warehouse.

The data in the warehouse are organized by multiple dimensions and put into a structure of dimensional modeling [1,7]. Dimensional modeling consists of arrangements of data into fact tables and dimension tables. The fact table contains important numerical measures to the user, as well as the keys to the dimension tables, which in turn include numerical and descriptive attributes [1,7]. Spatial attributes can appear in the fact table if they are key numeric facts for users, but are more commonly put in dimension tables, where they can provide numeric and descriptive information, including geographic ones.

Two well-known types of dimensional models are the *star schema* and *snowflake schema* [6]. An example of a star schema, shown in Fig. 2, gives information on fast food sales and locations. The fact table contains the keys to dimension tables and numeric attributes on total sales and total managers. The location dimension table gives, for each store, five geographic locations, ranging from county down to census block. They can be used for thematic mapping. Exact point locations of stores (X-Y coordinates)



Data Warehouses and GIS, Figure 1 The data warehouse and its data flows, spatial functions, and GIS components



Data Warehouses and GIS, Figure 2 Data warehouse star schema, with location included

could also be included, if deemed important enough. The other dimension tables provide information on store sales, products, and periodic reports.

GIS and spatial features can be present at several steps in the data warehouse, shown in Fig. 1. The operational databases may be spatially-enabled, so the data are geocoded prior to the ETL process. Location can be added as an attribute in the ETL step. Within the data warehouse, the fact tables or dimension tables may identify spatial units, such as ZIP code or county. The spatially-enabled tables may have address or X-Y coordinates. Geographical attributes would be located in the fact versus dimension table(s) if location rose to high importance for the user. For example, in a data warehouse of marketing data for sales regions, the region-ID is in the fact table.

GIS functionality is usually present in many of the analysis and modeling software packages shown on the right of Fig. 1.

GIS. The most powerful functionality would be in a full-featured GIS software package such as ArcGIS or GeoMedia, which can perform a wide variety of GIS functions from overlays or distance measurement up to advanced features such as geostatistics, modeling, 3-D visualization, and multimedia. It can enrich the uses of data warehouse information for utilities infrastructure; energy exploration, production, and distribution; traffic accident analysis; large scale auto insurance risk analysis; management of fleets of vehicles; and business intelligence for decision-making [8,9]. GIS and data warehouses often serve as parts of an organization's enterprise architecture. They can function in a collaborative, coupled environment with the other enterprise applications. A challenge is to connect separate enterprise software packages together through robust and efficient plug-in and connector software.

Online analytical processing (OLAP) is a set of rules for accessing and processing multidimensional data in the data warehouse. OLAP rules are focused on simple business decision-making that directly accesses the dimensions of data in the warehouse rather than on complex models. Among the main types of analysis are: (1) slice and dice, i. e. to divide complex datasets into smaller dimensions, (2) drill down, to seek more detail in a report, (3) what-if changes for single or multiple dimensions, and (4) access to the static, time-slice stores in the data warehouse [1]. OLAP is good at answering "why" and "what if" questions.

Specifically, OLAP refers the following characteristics of the information in the data warehouse [3]: (1) viewable in multiple dimensions, (2) transparent to the user, (3) accessible, (4) consistent in its reporting, (5) based on client/server architecture, (6) generic in dimensionality, (7) handling of dynamic sparse matrices, (8) concurrent support for multi-users, (9) cross-dimensional operations, (10) intuitive data manipulation, (11) flexible reporting, and (12) aggregation possible.

Spatial data can be integrated into the OLAP model, which is termed the *SOLAP model* [10,11,12]. The aggregation features of OLAP are modified for SOLAP to handle geographic attributes. One approach is to modify the OLAP's multidimensional data model "to support complex objects as measures, interdependent attributes for measures and aggregation functions, use of *ad hoc* aggregation functions and n-to-n relations between fact and dimension" [10].

SOLAP models are still under development, in particular to formulate improved SOLAP-based operators for spatial analysis, and more elaborate working prototypes [10]. In the future, a standard accepted SOLAP model would allow OLAP's what-if efficiencies for quick and flexible access to multidimensional data in data warehouse to include the complexity of spatial objects such as points, lines, and polygons. For some applications, such a model might eliminate the need for standard GIS software packages.

Business intelligence. Business intelligence (BI) software packages often have spatial features. BI consists of interactive models that are designed to assist decision-makers [1]. In the context of data warehouses, BI can conduct modeling based on information from the data warehouse for forecasting, simulations, optimizations, and economic modeling. Spatial capabilities can be present that include location in the modeling and produce results as maps.

Data mining. It seeks to reveal useful and often novel patterns and relationships in the raw and summarized data in the warehouse in order to solve business problems. The answers are not pre-determined but often discovered through exploratory methods [1]. The variety of data mining methods include intelligent agents, expert systems, fuzzy logic, neural networks, exploratory data analysis, and data visualization [1,13]. The methods are able to intensively explore large amounts data for patterns and relationships, and to identify potential answers to complex business problems. Some of the areas of application are risk analysis, quality control, and fraud detection.

There are several ways GIS and spatial techniques can be incorporated in data mining. Before the data mining occurs, the data warehouse can be spatially partitioned, so the data mining is selectively applied to certain geographies. During the data mining process, algorithms can be modified to incorporate spatial methods. For instance, correlations can be adjusted for spatial autocorrelation (or correlation across space and time), and cluster analysis can add spatial indices [14]. After data mining, patterns and relationships identified in the data can be mapped with GIS software.

Physical structure. Underneath the data warehouse's conceptual structure, data are physically stored either in multidimensional databases keyed to OLAP or in standard relational data-bases, which have slower performance. The biggest vendors for physical data warehouses are Oracle, IBM, and Microsoft. Some of their products have built-in spatial functionality, like Oracle Spatial 10g [15].

Large data warehouses may store many terabytes of information, cost several million dollars, and take up to two or three years to implement [1]. Their pluses include better decision-making capability, faster access, retention of data for longer time periods, and enhanced data quality [15]. Data quality is scrutinized and improved as part of the ETL process.

Spatially-enabled commercial data warehouses. Several major database and ERP vendors, including Oracle, IBM, and SAP, offer spatially-enabled database or data warehouse products. They are full-scale relational databases or data warehouses that have spatial functionality built into them, including spatial layers, geocoding, coordinate systems and projections, and spatial analysis techniques. Although the functionality is not as elaborate as the leading GIS software, it has gained in capability, to a level that satisfies many everyday business needs for maps and spatial processing.

This following discussion introduces essentials on commercial spatially-enabled data warehouses focusing on the Oracle Spatial 10g product. Oracle Spatial 10g supports the Oracle Data Warehouse, and has low-level to mid-level GIS functionality. The data warehouse is available in Oracle Spatial 10g and has features that include a multidimensional OLAP engine and built-in ETL features. This emerging trend demonstrates how existing mainstream enterprise packages can be commercially modified for mid-level spatial applications, without the necessity of connecting to traditional GIS software.

Design and construction of applications can be done through Oracle Warehouse Builder, a graphical design tool having: (1) a graphical "design environment" to create the data warehouse based on metadata, and (2) a "runtime environment," to convert the design into the physical processes that run the data warehouse [4]. For viewing, Oracle Spatial 10g's low-level "Map Viewer" provides simple maps of the dimensional attributes and summary data in the data warehouse [5]. For higher-level spatial analysis, major GIS vendor software such as ESRI's ArcGIS or Integraph's GeoMedia can be applied.

Key Applications

Data warehouses and GIS are applied to large-scale data sets for analysis of complex spatial problems that can include time. Important applications are for market segmentation, insurance analytics, complex urban transport, city traffic patterns and trends, patterns of credit results for regions or nations, international tourism consumer patterns, financial fraud, consumer loans, and matching census variables between bordering nations [1,8,9,16]. In this section, two examples are given of real-world applications: (1) auto insurance applications [10] and (2) traffic patterns for the city area of Portland, Oregon, over a time span of almost two decades [8].

Example of an auto insurance application. Spatial data warehouses can be built for large-scale analysis of auto insurance. In this example, the data warehouse resides in Oracle Spatial 10g. The business items in the data warehouse have location attributes that include census blocks, locations of policies, business sites, landmarks, elevation, and traffic characteristics. For data warehouses in auto risk insurance, maps can be produced that take spatial views from the usual ZIP code geography down to hundreds of small areas within the ZIPs [10]. This allows underwriters to set more refined policy pricing. The geoprocessing needs to be fast, many 10s of millions of location data processed per day [10].

Example of a local government application: City of Portland. The City of Portland illustrates use of a customized connector program to connect a data warehouse to a GIS. The data consist of city and regional traffic accidents from the Oregon Department of Transportation. The solution combined an SQL Server data warehouse with a customized program written in ArcObjects API (application programming interface) from ESRI Inc. There is a predefined schema of non-spatial and spatial attributes for transport of data between the data warehouse and the ArcObjects program.

The city's spatial data warehouse for city and regional traffic accidents has over fifteen years of data and fourteen dimensions, including time, streets, age and gender of participants, cause, surface, and weather. Following cleaning of data entering the data warehouse, location coordinates are added by the GIS team for each traffic accident. At the staging server, ETL extracts data weekly to two powerful clustered production servers. When updates are called for, the data warehouse repeats the ETL process, to output a new time slice of the data.

The volume of data is huge, so attention was given to mitigating performance bottlenecks [8]. The solution included optimizing replication of a time-slice of data, and partitioning the data warehouse cube to speed up the average access time. Users of the city's system utilize interactive maps of all the accident locations during the past decade and half, to supplement accident reports and give added insight for decisions [8]. The data are stored in an SQL Server data warehouse.

The customized program allows the GIS software to utilize part or all of the data warehouse. The City of Portland assigned programmers from its Corporate Geographic Information System (CGIS) Group to program the ETL and access modules, based on ArcObjects API for ArcGIS from ESRI [8]. Because of the scope of the programs, CGIS limited the types of questions users can ask to a predefined set. The accident outputs consist of tables and maps that are viewable on the Web. Having both query of tables and visualization of maps is regarded as crucial for users.

The benefits of this data warehouse/GIS approach include halving of replication time for a time slice of data, fast spatial queries, and response times shortened by 20-fold or more [8].

Future Directions

The contributions of GIS and spatial technologies to data warehouse applications are to provide mapping and spatial analysis. Data warehouse applications can recognize locations of organizational entities such as customers, facilities, assets, facility sites, and transport vehicles. GIS provides visualization and exploration benefits to understand patterns and relationships of enterprise information in the data warehouse, and support better decisions. The challenges are to design spatially-enabled data warehouse architectures that provide added value to corporate users and customers, are flexible enough to change with the rapid technology advances in this field, and are efficient enough to achieve satisfactory throughput and response times.

Future advances are anticipated that make data warehouses more efficient, improve integration of data warehouses with GIS, tune the analytic outputs to the typical data-warehouse users, and coordinate the spatial data warehouses with other enterprise software such as ERP, supply chain, and customer relationship management (CRM). Since the large data sets for each time-slice are written permanently to the data warehouse, the location coordinates are often added in the ETL stage or earlier. Future software needs to have more efficient tools available during ETL such as geocoding, to minimize employee time spent. There needs to be faster and more seamless connector software and interfaces between commercial data warehouse and GIS software. Analytic software such as SAS and ArcGIS need to have analysis capabilities that coordinate better with the data warehouse. The future shows promise that SOLAP will become standardized as an extension of OLAP. Spatial data warehouses serve analytic users who are interested in patterns, associations, and longitudinal trends. In the future, the GIS and spatial functions will become even better focused on serving these users.

Cross References

► OLAP, Spatial

Recommended Reading

1. Gray, P.: Manager's guide to making decisions about information systems. John Wiley and Sons, New York (2006)
2. Inmon, W.H.: Building the data warehouse. John Wiley and Sons, New York (1990)
3. Codd, E.F.: Twelve rules for on-line analytic processing. Comput. (1995)
4. Rittman, M.: An introduction to Oracle warehouse builder 10g. DBAzine.com. August 19. Available at www.dbazine.com (2006)
5. Rittman, M.: GIS-enabling your Oracle data warehouse. DBAzine.com. April 18. Available at www.dbazine.com (2006)
6. Gray, P., Watson, H.: Decision support in the data warehouse. Prentice Hall, Upper Saddle River (1998)
7. Jukic, N.: Modeling strategies and alternatives for data warehousing projects. Commun. ACM. **49**(4), 83–88 (2006)
8. SQL Server Magazine: City of Portland Tames Massive SQL Server Data Warehouse. June 27. Available at www.sqlmag.com (2002)
9. Reid, Hal: Applying Oracle Spatial to a very large insurance problem. Location Intelligence, June 20. Available at www.locationintelligence.net (2006)
10. Bimonte, S., Tchounikine, A., Miquel, M. Towards a spatial multidimensional model. In Proceedings of the 8th ACM International Workshop on Data Warehousing and OLAP, November 4–5, Bremen Germany, pp. 39–46 (2005)
11. Malinowski, E., Zimanyi, E.: Representing spatiality in a conceptual multidimensional model. In Proceedings of the 12th annual ACM international workshop on geographic information systems. ACM Press, Washington DC, pp. 12–22 (2003)
12. Marchand, P., Brisebois, A., Bedard, Y., Edwards, G.: Implementation and evaluation of a hypercube-based method for spatiotemporal exploration and analysis. J. Int. Soc. Photogrammetry Remote Sensing **59**,6–20 (2003)
13. Tan, P-N, Steinbach, M., Kumar, V.: Introduction to data mining. Addison-Wesley, Upper Saddle River (2005)
14. Viswanathan, N., Pick, J.B., Hettrick, W.J., Ellsworth, E.: An analysis of commonality in the twin metropolitan areas of San Diego, California and Tijuana, Mexico, J. Soc. Plann. Sci. **39**, 57–79 (2005)
15. Oracle: Oracle Data Warehousing. Available at www.oracle.com (2006)
16. Pick, J. B., Viswanathan, N., Hettrick, W. J.: A dual census geographical information system in the context of data warehousing. Proceedings of the Americas Conference on Information Systems. Association for Information Systems, Atlanta, Georgia, pp. 265–278 (2000)

Data Warehousing

► Database Schema Integration

Database Indexing

► Indexing, Hilbert R-tree, Spatial Indexing, Multimedia Indexing

Database Integration

► Ontology-Based Geospatial Data Integration

Database Management

► Smallworld Software Suite

Database Schema Integration

RACHEL POTTINGER
Department of Computer Science,
University of British Columbia, Vancouver, BC, Canada

Synonyms

Resolving semantic schema heterogeneity; Schema mapping; Data warehousing; Peer data management

Definition

Data's organization is referred to as a *schema*. When multiple sources of data must be combined to retrieve information that is not contained entirely in either one, typically they do not have the same schemas. For example, database A's schema may store information about roads as "roads" and database B's schema may use "streets" for roads. In order for information from database A and database B to be integrated, they must resolve the fact that the same information is stored in different schemas; this is referred to as *semantic schema heterogeneity*. To resolve semantic schema heterogeneity, there must be some mechanism to allow queries to be asked over multiple schemas. This involves (1) creating a database schema that is the integration of the original schemas (i. e., performing *database schema integration*), (2) creating a *schema mapping* between the original schemas (a process known as *schema matching*), and (3) having a system that allows the mappings to be used to translate queries.

Historical Background

Data stored in a database are typically curated or owned by one organization. This organization controls not only what data can be entered into the database, but how that data is organized as the *schema* of the database. In a relational database, a schema is made up of relations—descriptions of a concept—that are composed of single-valued text attributes. For example a university mailing database ("Mail") might include the relation of buildings (Table 1). This relation, named Building has attributes of Address, Location, Department, and has two instances.

Database Schema Integration, Table 1 An example relation named "Building" which represents building locations in a mailing database for a university

Address	Position	Department
1984 West Mall	49°15′57″N 123°15′22″W	Geography
2366 Main Mall	49°15′40″N 123°14′56″W	Computer Science

Database Schema Integration, Table 2 A representation of a building called "Bldg" in a maintenance database for a university

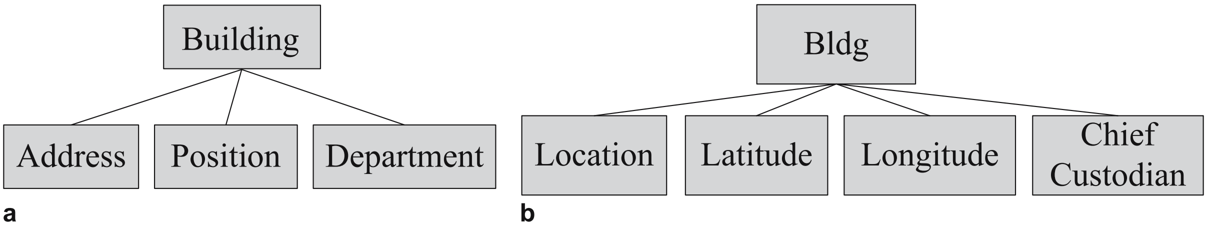
Location	Latitude	Longitude	Chief Custodian
1984 West Mall	49°15′57″N	123°15′22″W	Pat Smith
2366 Main Mall	49°15′40″N	123°14′56″W	Chin Yu

When each database is treated in isolation, there is a unique representation of concepts in the schema. For example, the address of a building in Building is represented by an attribute "Address". To find all building addresses, a query would have to be asked to find all "Address" attributes of the Building relation. However, different databases may represent their schemas in different ways for same concepts. Table 2 shows how a building might be differently represented in a maintenance database ("Maintenance"): Each contains separate information, but combining the databases would allow new information to be discovered that cannot be found by using each database separately. For example, given that the Geography Department is found at 1984 West Mall, and the Chief Custodian for 1984 West Mall is Pat Smith, by combining the information from the two sources, an administrator could tell that Pat Smith is the Chief Custodian for the Geography department. Querying multiple relations simultaneously requires understanding the differences in the schemas. For example, querying all building addresses now requires finding all "Address" attribute values in Building (Table 1) and all "Location" attribute values in Bldg (Table 2). Moreover, the schemas differ in more complicated ways as well: the concept of "Position" in Building corresponds to the concatenation of the "Latitude" and "Longitude" attributes in Bldg. The difference between schemas is known as semantic heterogeneity. Naturally, semantic heterogeneity was raised as a concern as soon as databases that were created needed to be combined, and the need for database schema integration was recognized very early on [1].

Scientific Fundamentals

Schema Mapping Creation

Before the schemas can be integrated, the similarities between attributes must be determined so that a mapping can be created between the schemas. For example, given the relations in Table 1 and Table 2, it is obvious to a human that "Position" in Building (Table 1) corresponds to the concatenation of "Latitude" and "Longitude" in Bldg (Table 2). For scalability, this correspondence (mapping) needs to be discovered in a more automatic fashion. However, because determining if two schema elements are the same is inherently reliant on information that is



Database Schema Integration, Figure 1 A graphical representation of **a** the Building relation from the Mail database (Table 1) and **b** the Bldg relation from the Maintenance database (Table 2)

only present in the designers' heads, any semiautomatic approach must defer the ultimate decision about when two concepts are equal to a human being. Thus, the goal of *schema matching* is to create computer tools to leverage a person's decisions to make the best possible mappings. A survey of the techniques can be found in [2]. Information about how the schemas are related—which is necessary in schema matching—can be found from a number of different sources of information:

- Schema:information about the schema, particularly:
  - Name: the name of an element in a schema (e. g., "Building")
  - Type: the type of a schema element (e. g., "Integer")
  - Structure: information about relationships between elements in a schema (e. g., the fact that "Building" contains an attribute called "Address")
- Data: the data in database instances can provide additional evidence about how the schemas are related. For example, the data values may be identical or have similar formats between instances in different schemas e. g., the data values for the "Address" attribute in Table 1 are more similar to those of the "Location" attribute in Table 2 than the "Position" attribute values. This suggests that "Location" is more similar to "Address" than "Position".

Most schema matching algorithms treat schemas as graphs. Figure 1a and b depict the graphs of the Building relation in Table 1 and the Bldg relation in Table 2 respectively.

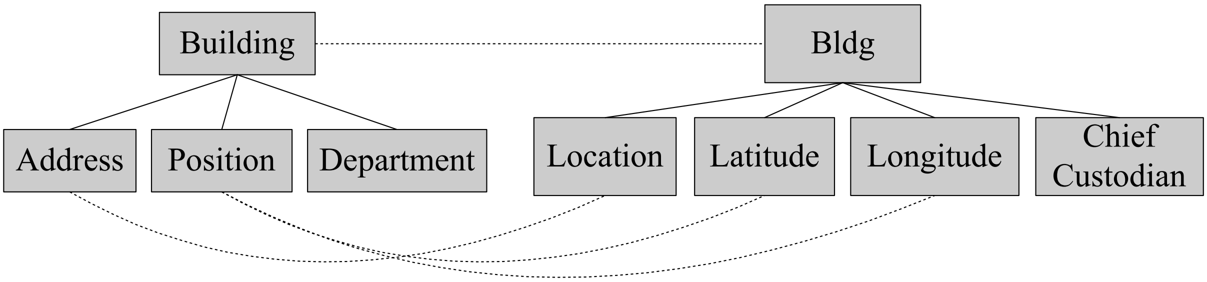
A typical schema matching algorithm might use the following procedure to account for the example in Fig. 1:

- Building matches Bldg because "Bldg" is an abbreviation of "Building", and they each contain a similar number of subelements.
- Address and Position both contain names with similar meanings to Location, and both are children of an already matched element. Both elements are of type *string*, which does not resolve whether Location is more similar to Address or Position. However, looking at the data instances, the data values of Address and Location are identical and have similar form. Given all of this structural, type, element name, and data value information, Address and Location would likely be matched together.
- The initial consideration for matching Position would either be to Location, Latitude, or Longitude, which is based on the structural information (i. e., they are all children of matched elements) and the name of the elements. Just as in the previous step, where the data values helped to decide that Address should be mapped to Location, the data instances reduce some of the ambiguity of what Position should be mapped to; the values of Position are more similar to those of Latitude and Longitude. This is reinforced by the information that the match between Address and Location is strong; it weakens the chances that Location corresponds to Position. Hence a matching algorithm is likely to conclude that Position corresponds to either (1) Latitude (2) Longitude or (3) the concatenation of Latitude and Longitude. The third option, while obviously the correct one, is difficult for systems to create, since considering matches that are more complex than one-to-one matches adds substantially to the problem.
- Based on the fact that Department and Chief Custodian are both attributes of relations that have been previously matched (Building and Bldg., respectively), both Department and Chief Custodian would be checked to see if matches existed for them as well. However, given that there are no elements with names of similar meanings and their instances are also very dissimilar, they will likely not be matched to any element.

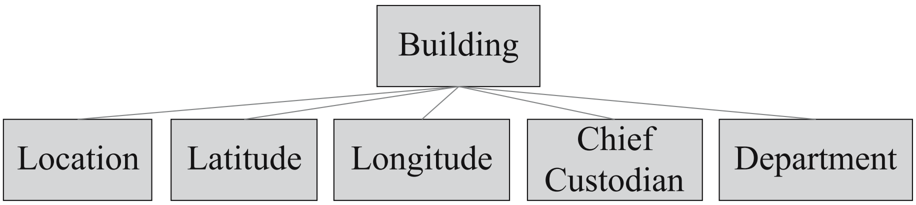
Thus, the likely mapping between the two schemas is as shown in Fig. 2.

Taking all the schema and data information into account is a complicated procedure, and many algorithms have been studied. Some representative algorithms are:

- Learning source descriptions (LSD) [3] is a system that creates mappings using machine learning methods (see [4] for a reference) as follows. A user creates some matches by hand (e. g., "Building" matches "Bldg"). These mappings are then used to train various learners (e. g., the name, type, structure, and data match-



Database Schema Integration, Figure 2 A likely mapping between the elements in Fig. 1



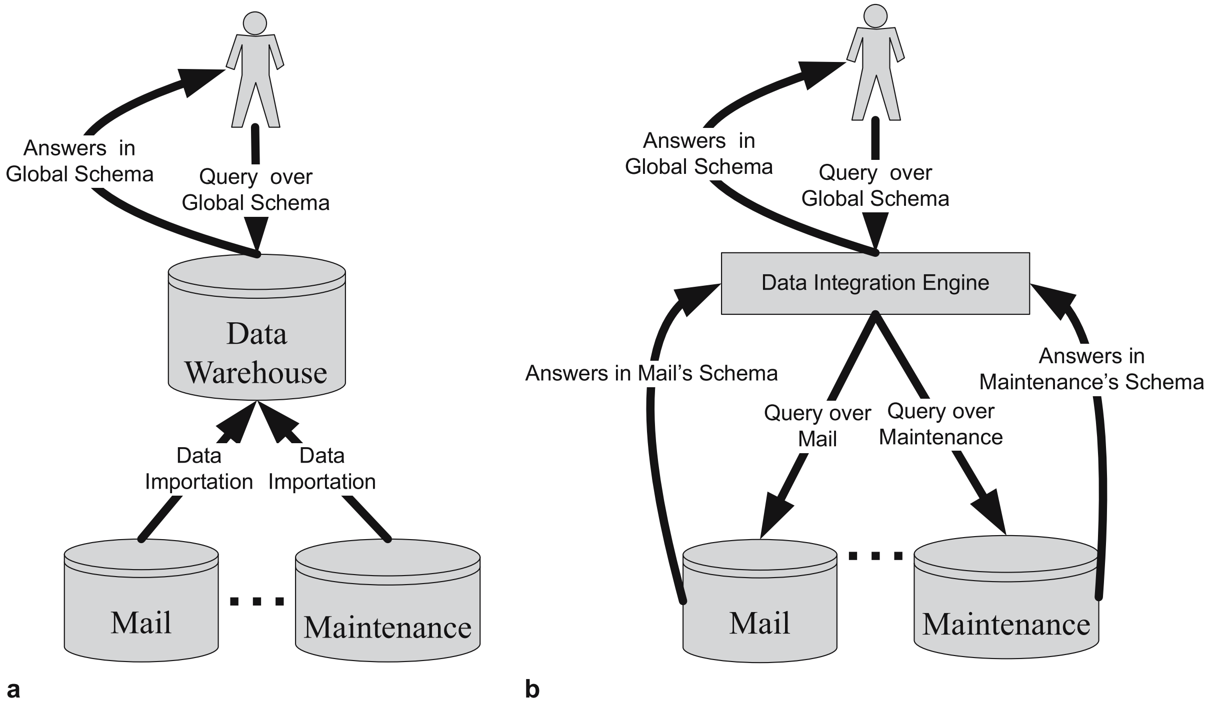
Database Schema Integration, Figure 3 A possible global schema created from the input schemas in Fig. 1

ers) so that they will recognize aspects of the input, for example, in one case to look for abbreviation. Then a meta-learner is trained to recognize the combination of those learners as being a valid match, for example, in this case, to weigh the name matcher highly. Given the information provided by the meta-learner, LSD can generalize the small number of matches created by user input into many more matches.

- Similarity flooding [5]; here, an initial mapping is created by checking for similarity in names, type, and data. For example, in the schemas in Fig. 1, the algorithm may discover that Building in Table 1 matches Building in Table 2, and that Position matches Location. Next, the key notion of the graph-based Similarity Flooding method is applied: if elements are mapped to each other, then schema elements that connect to them are more likely to be mapped to each other. For example, initially Position may be ranked as fairly unrelated to any attribute. However, given that Building matches Bldg, and Position is an attribute of Building, as Latitude and Longitude are attributes of Bldg, then Position is more likely similar to Latitude and Longitude. This similarity is then fed back to create new matches until no new matches can be found.
- Clio [6] starts where LSD and similarity flooding ends. In Clio, the input is a set of simple correspondences, like the ones in Fig. 2. However, these simple correspondences are not complex enough to explain how to translate data from one schema to the other. For example, in Fig. 2, Position is clearly related to Latitude and Longitude, but it is not clear that one must concatenate a Latitude and Longitude value to produce a Position. Clio uses similar techniques to those used to find the correspondences (i. e., examining the schema and data information) and provide the details of how to translate an instance from one to the other. The output of Clio is a query in either SQL (Structured Query Language—the standard query language for relational data), or for XML (Extensible Markup Language—a semi-structured data representation) an XML Query. This query can translate the data from one source to another.

Key Applications

After the schema mapping has been formed, the correspondences between schema elements exist, but it still remains to enable queries to be simultaneously asked over all the input database schemas. For example, Fig. 2 shows how the two schemas in Table 1 and Table 2 are related, but does not allow them to be queried simultaneously. There are a number of different mechanisms for this; we concentrate on three of them: data warehousing, database schema integration, and peer data management systems. In both data warehousing and database schema integration, a single global schema is created, and queries are asked over that schema. For example, Fig. 3 shows an example of a global schema that might be created as a result of the mapping (Fig. 2). After such a mapping is created, a global schema is generally made by combining the elements in the source schemas, and removing duplicates, which can either be done by hand, or through some more automatic method. Batini, Lenzerini, and Navathe [7] provide a thorough explanation of what considerations have to be made, and also provide a survey of some early work. Additional work



Database Schema Integration, Figure 4 Data Warehousing (**a**) and Database schema integration (**b**) both involve global schemas, but in data warehousing, the data is imported to a central location, and in database schema integration, the data is left at a source, and each query requires retrieving data from the sources

on this problem can be found in other sources, including [8,9,10].

However, aside from having a single schema in which the sources are queried, the architecture of data warehousing and database schema integration systems differs in a key way: in a data warehousing situation the data is imported into a single store (Fig. 4a); and in a database schema integration system, the data remains in the sources, and at query time the user's query over the global schema is rewritten into a set of queries over the source schemas (Fig. 4b).

Data Warehousing

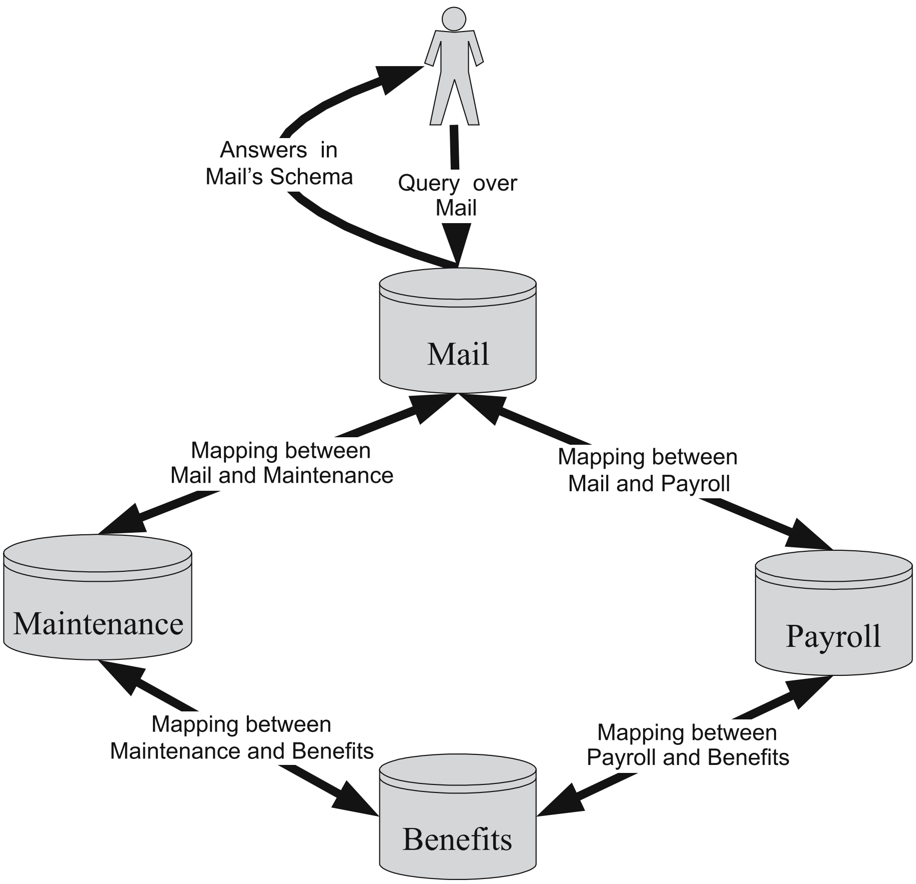
In data warehousing (Fig. 4a), the data is imported from the sources. For this to happen, a global schema needs to be created for the source data to be imported into. This schema can be created through a process such as the one described to create the schema in Fig. 3. An example of such a system can be found in [11]. Additional complications for creating the global schema include that the warehouse is often used for answering different types of queries, and must be optimized for this. For example, the data cube [12] creates a schema that allows for easy access to such information as categorizing data on road length by national, region, and city.

Database Schema Integration

In a database schema integration system, as shown in Fig. 4b, the data are maintained separately at each source. At query time each query over the global schema has to be reformulated into a query over a local source (see [13] for a survey).

Peer Data Management Systems

A peer-to-peer (P2P) system is a loose confederation of sources, such as the one shown in Fig. 5, where each peer may contain different data, and is free to come and go at any time. Typical P2P networks such as Napster have no schemas, but instead issue blanket queries, e. g., "Britney Spears". In GIS and other semantically rich situations, this is insufficient, leading to the emerging trend of peer data management systems (PDMSs) (e. g., [14,15]). PDMSs maintain the flexibility of a P2P system, but allow queries to be asked over schemas rather than simple existence queries. This is in contrast to database schema integration or data warehousing systems, since the networks are so flexible, and there is no overriding authority. Hence, rather than having a centralized mechanism for ensuring that the schemas are the same, or creating a single, global schema, typically each source has its own schema. Then,



Database Schema Integration, Figure 5 In a Peer Data Management System, an ad-hoc network of peers exists. Queries are typically asked over the local peer's schema, and mappings between sources are used to translate queries and mappings back and forth

each new peer that enters the system must create a mapping to one or more peers already in the network. Peers for which a direct mapping exists between are called acquaintances. For example, if the Mail database were just entering the network, it might create mappings between itself and the Maintenance and Payroll databases, which would then become its acquaintances.

At query time, users ask queries over their local sources in one peer. Queries are then translated into queries over each acquaintance, which passes back its data, and then also forwards the query to its acquaintances that respond with their answers as well. Eventually, all relevant data are passed back to the peer that initiated the query. For example, a user of the Mail database query for all building positions would query for "Position" in Mail. In addition to those answers being returned, the PDMS would also look at the mappings to Maintenance and Payroll and return the concatenation of Latitudes and Longitudes in Maintenance, and whatever corresponded to Location in Payroll. Additionally, Maintenance would then check its mapping to Benefits to see if any corresponding information existed in it, before all answers were returned. This reliance on the ability of the peers to compose mappings between sources, is unfortunately a difficult problem that is the object of ongoing research (e. g., [16,17]).

Future Directions

Currently, as indicated, most research on schema mapping is on one-to-one schema mappings, and solving the simple problem. More research (e. g., [18]) is focusing on having mappings that are more complex than one-to-one (e. g., the mapping of Position to the concatenation of Latitude and Longitude in Fig. 2). Current work on data integration is also concentrating on how to combine data from sources that are structured (e. g., have a relational structure) with those who have minimal structure (e. g., plain text), and how to allow more structured querying of loose confederations of data [19].

Cross References

► Conflation of Geospatial Data
► Geocollaboration
► Geospatial Semantic Integration
► Geospatial Semantic Web, Interoperability
► Metadata and Interoperability, Geospatial

▶ Mobile P2P Databases
▶ Ontology-Based Geospatial Data Integration

Recommended Reading

1. Shu, N.C., Housel, B.C., Lum, V.Y.: CONVERT: A high level translation definition language for data conversion. Commun. ACM **18**, 557–567 (1975)
2. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB J. **10**, 334–350 (2001)
3. Doan, A., Domingos, P., Halevy, A.: Reconciling schemas of disparate data sources: a machine learning approach. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD) 2001, May 21–24. Santa Barbara, CA
4. Mitchell, T.M.: Machine Learning, 1st edn. McGraw-Hill, New York: (1997)
5. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity flooding: a versitile graph matching algorithm and its application to schema matching. In: Proceedings of the International Conference on Data Engineering 2002, June 3–6, Madison, WI
6. Miller, R.J., Haas, L.M., Hernández, M.A.: Schema mapping as query discovery. In: Proceedings of the Very Large Data Bases Conference (VLDB) 2000, Sept. 10–14, Cairo, Eqypt
7. Batini, C., Lenzerini, M., Navathe, S.B.: A comparative analysis of methodologies for database schema integration, ACM Comput. Surv. **18**, 323–364 (1986)
8. Buneman, P., Davidson, S.B., Kosky, A.: Theoretical aspects of schema merging. In: International Conference on Extending Database Technology (EDBT) 1992, March 23–27, Vienna, Austria
9. Melnik, S., Rahm, E., Bernstein, P.A.: Rondo: A programming platform for generic model management. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD) 2003, June 9–12, San Diego, CA
10. Pottinger, R.A., Bernstein, P.A.: Merging models based on given correspondences. In: Proceedings of the Very Large Data Bases Conference (VLDB) 2000, Sept. 10–14, Cairo, Egypt
11. Calvanese, D., Giacomo, G.D., Lenzerini, M., Nardi, D., Rosati: Schema and Data Integration Methodology for DWQ. Technical report, DWQ Consortium. TR# DWQ-UNIROMA-004, Sept. 1998, Università di Roma "La Sapienza" (1998)
12. Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., Pirahesh, H.: Data Cube: A relational aggregation operator generalizing group-by, cross-tab, and sub totals. Data Min. Knowl. Disc. **1**, 29–53 (1997)
13. Ullman, J.D.: Information integration using logical views. In: Proceedings of the International Conference on Database Theory (ICDT) 1997, January 8–10, Delphi, Greece
14. Bernstein, P.A., Giunchiglia, F., Kementsietsidis, A., Mylopoulos, J., Serafini, L., Zaihrayeu, I.: Data management for peer-to-peer computing: a vision. In: International Workshop on the Web and Databases (WebDB) 2002, June 6–7, Madison, WI
15. Halevy, A.Y., Ives, Z.G., Suciu, D., Tatarinov, I.: Piazza: Data management infrastructure for semantic web applications. In: Proceedings of the International Conference on Data Engineering (ICDE) 2003, March 5–8, Bangalore, India
16. Fagin, R., Kolatis, P.G., Popa, L., Tan, W.C.: Composing schema mappings: second-order dependencies to the rescue: In: Symposium on Principles of Database Systems (PODS) 2004, June 14–16, Paris, France
17. Madhavan, J., Halevy, A.Y.: Composing mappings among data sources. In: Proceedings of the Very Large Data Bases Conference (VLDB) 2003, Sept. 9–12, Berlin, Germany
18. Dhamankar, R., Lee, Y., Doan, A., Halevy, A.Y., Domingos, P.: iMAP: discovering complex mappings between database schemas. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), 2004, June 13–18, Paris, France
19. Madhavan, J., Halevy, A.Y., Cohen, S., Dong, X.L., Jeffrey, S.R., Ko, D., Yu, C.: Structured data meets the web: a few observations. IEEE Data Eng. Bull. **29**, 19–26 (2006)

Databases, Relational

▶ Constraint Databases, Spatial

Datalog, SQL

▶ Constraint Database Queries

Data-Structure, Spatial

▶ Quadtree and Octree

Data-Structures

▶ Indexing Schemes for Multi-dimensional Moving Objects

Daytime Population

▶ Population Distribution During the Day

DE-9IM

▶ Dimensionally Extended Nine-Intersection Model (DE-9IM)

Dead-Reckoning

▶ Moving Object Uncertainty

Decision Rules

► Multicriteria Decision Making, Spatial

Decision Support

► Rural Hydrologic Decision Support

Decision Support Systems

MARTIN D. CROSSLAND
School of Business, Oral Roberts University,
Tulsa, OK, USA

Synonyms

Knowledge based systems

Definition

A *decision support system* (DSS) is a computer-based system that combines data and decision logic as a tool for assisting a human decision-maker. It usually includes a user interface for communicating with the decision-maker. A DSS does not actually make a decision, but instead assists the human decision-maker by analyzing data and presenting processed information in a form that is friendly to the decision-maker.

Main Text

Decision systems are typically combination of rule sets and decision logic (a "decision engine") that operate on particular data contained within a specified database. The data in the database are processed and arranged in such a way to be accessible to the decision engine. The decision engine may aggregate various data to form usable information for the decision-maker, or it may search the data to find meaningful patterns that may also be useful.

Decision support systems (DSS) are increasingly being combined with *geographic information systems* (GIS) to form a hybrid type of decision support tool known as a *spatial decision support system* (SDSS). Such systems combine the data and logic of a DSS with the powerful spatial referencing and spatial analytic capabilities of a GIS to form a new system that is even more valuable than the sum of its parts.

Cross References

► Spatial Decision Support System

Decision Support Tools for Emergency Evacuations

► Emergency Evacuation, Dynamic Transportation Models

Decision-Making Effectiveness with GIS

MARTIN D. CROSSLAND
School of Business, Oral Roberts University,
Tulsa, OK, USA

Synonyms

Marketing information system; Business application

Definition

A *spatial decision support system* (SDSS) is a computer-based system that combines conventional data, spatially-referenced data and information, and decision logic as a tool for assisting a human decision-maker. It usually includes a user interface for communicating with the decision-maker. It is the logical marriage of geographic information systems (GIS) and decision support systems (DSS) technologies to form an even more powerful assistant for decision-making. In these systems the database and decision engine of the DSS is enhanced with the capability of specifying, analyzing, and presenting the "where" of information sets within the GIS. An SDSS does not actually make a decision, but instead assists the human decision-maker in reviewing and analyzing data and presenting processed information in a form that is friendly to the decision-maker. Effectiveness of an SDSS generally concerns how much "better" a decision made is relative to a decision made without such technology support.

Historical Background

Geographic information systems (GIS) have been increasingly employed to the tasks of modern problem-solving. It has been recognized that many everyday situations and problems involve information and data that contain spatially-referenced features or components. For example, traffic problems happen in specific places, and those places have various spatially-referenced descriptors, such as shape of the roadway (i. e., curved versus straight), or how close the problems occur to an intersection with another roadway. Indeed, some of the most influential components of many types of business or societal decisions involve some determination or estimation of the nearness or even co-location of two or more features or items of interest. For these types of questions involving the exact location of such features, one may wish to employ computer-based tools which reference information databases to determine what information is relevant to the decision, as well as how to use that information.

The science of decision-making has employed computer-based or computer-enhanced methods for quite some time now. Some of the "new" automated methods of decision making were first presented over forty-five years ago [1]. The use of computers has since evolved from the early automation into quite complex and powerful *decision support systems* (DSS) that utilize various storehouses of information (e. g., databases), along with specified or derived rules sets. These assist or enable a user to make effective decisions about specific subject areas, or *knowledge domains*. The development, use, and analysis of DSS has been described quite extensively (e. g., in [2] and [3]).

Solutions for many types of problems have remained elusive, however, especially when the data involved spatial components, because deciding how such data could be related and analyzed was not straightforward. In a typical database, information records may be related based on textual or numerical fields that certain records hold in common among two or more tables in the database. For example, a table of employees' personal information might be linked to a table of total annual compensation by their social security number.

However, for many other types of information the only features certain entries in the database may have in common are their locations. For example, one might want to know something about whether there is some statistically significant relationship between nitrogen-enrichment in bodies of water and the application of manure-based fertilizers on crop lands. The answer to this type of question may involve not only the precise locations of the two data sets in question, but various other spatially-referenced information as well (e. g., direction of slope of the land, movements of rivers and streams, etc.). In addition, the only available means of relating these data sets may be their locations.

For solving these latter types of problems, the *spatial decision support system (SDSS)* has evolved. SDSS is the logical marriage of GIS and DSS technologies to form an even more powerful assistant for decision-making. In these systems the database and decision engine of the DSS is enhanced with the capability of specifying, analyzing, and presenting the "where" of information sets within the GIS. As early as over twenty-five years ago, a somewhat prophetic *Harvard Business Review* article [4] described several examples of problem types that should be solvable using GIS-like computer technologies. Later reports have actually named SDSS as the technology of choice [5] and [6].

Scientific Fundamentals

As decision support technologies emerged, a natural question was whether they actually provided the decision-maker with increased decision capacity, better accuracy, reduced times for decisions, and other measures of decision effectiveness. Even from early on, quite a number of studies were reported for the more generalized form of decision support systems [7] and [8]. As these studies progressed and GIS/SDSS emerged, it was soon realized that similar studies were needed regarding the effectiveness decision-making when using GIS and SDSS.

One of the earliest studies actually quantified SDSS decision effectiveness [9] and demonstrated that decision-makers had shorter decision times and higher accuracies for problems involving spatially-referenced information, even for problems of different complexity. These findings were confirmed and replicated in other studies [10] and [11].

Key Applications

Spatial decision support systems are now employed in many industries and civic applications, from business uses to public health management. Some interesting applications include the following.

Urban Construction

SDSS are used in analyzing, designing, and implementing urban constructions projects. They often include discussions of using *fuzzy set theory* for multicriteria decision making [12].

Modeling Natural Systems for Proper Management

One large application area is modeling natural systems for proper management, dealing with topics such as forest systems, ecosystems management, physical environmental modeling, petroleum waste management, and others. They involve activity planning as well as conservation topics [13] and [14].

Planning

A broad application area involves various types of planning decisions, including urban, environmental, and telecommunications infrastructure [15].

Agriculture and Land Use

Another large group of applications is focused primarily on agriculture and agricultural uses of land resources, including fertilization and nutrient management, and also crop and livestock systems and land use planning [16] and [17].

Group SDSS

An interesting set of early work in DSS included how individual stakeholders could combine their decision-making tasks in order to arrive at higher quality decisions as a group. There has been considerable interest in bringing that collaborative approach to group decision-making with SDSS [18].

Health Care Management

Health care is a widely-cited application area of SDSS for many types of analyses, all the way from disease outbreak studies to provision of public health care services [19] and [20].

Forestry Management and Conservation

A number of SDSS applications focus specifically on forestry management and natural resource conservation [21] and [22].

Traffic Management

SDSS has bee utilized in studies concerning traffic management systems, where vehicular traffic flows are documented and analyzed [23] and [24].

Marketing Information System

SDSS has been applied in what has been termed a marketing information system, where the decision maker can use spatially-referenced data in studying the four domains of the marketing decision universe: price, place, positioning, and promotion [25].

Environmental Hazards Management

Environmental hazards management includes some of the highest-profile application of SDSS – for predicting, planning for, and responding to various risks and hazards, both man-made and natural [26] and [27].

Water Resources Management

Water resources management applications focus on tracking locations, flows, quality, and sustainability of water resources [28] and [29].

Future Directions

GIS have been used increasingly in decision support roles. The resulting spatial decision support systems have been found to be valuable assets in many different arenas of decision-making, from business to public management to public resources management. Since almost everything on the earth has a “where” component that affects how one looks at it and evaluates it, a natural approach would be to use this spatially-referenced information whenever possible to help make timely, effective business decisions and life decisions.

Cross References

► Spatial Decision Support System

Recommended Reading

1. Simon, H.: The New Science of Management Decisions. Harper and Row, New York (1960)
2. Bonczek, R.H., Holsapple, C.W., Whinston, A.B.: Foundations of Decision Support Systems. Academic Press, Orlando, FL (1981)
3. Sprague, R.H.: Framework for DSS. MIS Quart. **4**(4), 1–26 (1980)
4. Takeuchi, H., Schmidt, A.H.: The new promise of computer graphics. Harvard Bus. Rev. **58**(1), 122–131 (1980)
5. Armstrong, A.P., Densham, P.J.: Database organization strategies for spatial decision support systems. Int. J. Geogr. Inf. Sys. **4**(1), 3–20 (1990)
6. Keenan, P.B.: Spatial Decision Support Systems. In: Mora, M., Forgionne, G., Gupta, J.N.D. (eds.) Decision Making Support Systems: Achievements and Challenges for the New Decade. pp. 28–39. Idea Group, Harrisburg, PA (2003)
7. Lucas, H.C.: Performance and the use of an information system. Manag. Sci. **21**(8), 908–919 (1979)
8. Money, A.T.D., Wegner, T.: The quantification of decision support within the context of value analysis. MIS Quart. **12**(2), 223–236 (1988)
9. Crossland, M.D.: Individual decision-maker performance with and without a geographic information system: an empirical study. Unpublished doctoral dissertation, Kelley School of Business. Indiana University, Bloomington, IN (1992)
10. Crossland, M.D., Wynne, B.E., Perkins, W.C.: Spatial decision support systems: an overview of technology and a test of efficacy. Decis. Supp. Sys. **14**(3), 219–235 (1995)
11. Mennecke, B.E., Crossland, M.D., Killingsworth, B.L.: Is a map more than a picture? The role of SDSS technology, subject characteristics, and problem complexity on map reading and problem solving. MIS Quart. **24**(4), 601–629 (2000)
12. Cheng, M.Y., Ko, C.H., Chang, C.H.: Computer-aided DSS for safety monitoring of geotechnical construction. Aut. Const. **11**(4), 375–390 (2002)
13. Karlsson, J., Ronnqvist, M., Frisk, M.: RoadOpt: A decision support system for road upgrading in forestry. Scand. J. Forest Res. **21**(7), 5–15 (2006)
14. Zhang, B.S., et al.: Predictive modeling of hill-pasture productivity: integration of a decision tree and a geographical information system. Agri. Sys. **87**(1), 1–17 (2006)
15. Culshaw, M.G., et al.: The role of web-based environmental information in urban planning – the environmental information system for planners. Sci. Total Env. **360**(1–3), 233–245 (2006)

16. Choi, J.Y., et al.: Utilizing web-based GIS and SDSS for hydrological land use change impact assessment. Trans. ASAE **48**(2), 815–822 (2005)
17. Ochola, W.O., Kerkides, P.: An integrated indicator-based spatial decision support system for land quality assessment in Kenya. Comp. Electron. Agric. **45**(1–3), 3–26 (2004)
18. Hendriks, P., Vriens, D.: From geographical information systems to spatial group decision support systems: a complex itinerary. Geogr. Env. Mod. **4**(1), 83–104 (2000)
19. Johnson, M.P.: Spatial decision support for assisted housing mobility counseling. Decis. Supp. Sys. **41**(1), 296–312 (2005)
20. Rushton, G.: Public health, GIS, and spatial analytic tools. Ann. Rev. Pub. Health **24**, 43–56 (2003)
21. Geneletti, D.: A GIS-based decision support system to identify nature conservation priorities in an alpine valley. Land Use Policy **21**(2), 149–160 (2004)
22. Rao, K., Kumar, D.S.: Spatial decision support system for watershed management. Water Res. Manag. **18**(5), 407–423 (2004)
23. Randall, T.A., Churchill, C.J., Baetz, B.W.: Geographic information system (GIS) based decision support for neighborhood traffic calming. Can. J. Civ. Eng. **32**(1), 86–98 (2005)
24. Tarantilis, C.D., Kiranoudis, C.T.: Using a spatial decision support system for solving the vehicle routing problem. Inf. Manag. **39**(5), 359–375 (2002)
25. Hess, R.L., Rubin, R.S., West, L.A.: Geographic information systems as a marketing information system technology. Dec. Supp. Sys. **38**(2), 197–212 (2004)
26. Keramitsoglou, I., Cartalis, C., Kassomenos, P.: Decision support system for managing oil spill events. Env. Manag. **32**(2), 290–298 (2003)
27. Martin, P.H., et al.: Development of a GIS-based spill management information system. J. Hazard. Mat. **112**(3), 239–252 (2004)
28. Liu, C.W.: Decision support system for managing ground water resources in the Choushui River alluvial in Taiwan. J. Am. Water Res. Assoc. **40**(2), 431–442 (2004)
29. Nauta, T.A., Bongco, A.E., Santos-Borja, A.C.: Set-up of a decision support system to support sustainable development of the Laguna de Bay, Philippines. Mar. Poll. Bull. **47**(1–6), 211–219 (2003)

Decision-Making, Multi-Attribute

► Multicriteria Decision Making, Spatial

Decision-Making, Multi-Criteria

► Multicriteria Decision Making, Spatial

Decision-Making, Multi-Objective

► Multicriteria Decision Making, Spatial

deegree Free Software

MARKUS LUPP
lat/lon GmbH, Bonn, Germany

Synonyms

deegree open source framework; deegree library; Java; Public-domain software; Open source; SDI (Spatial data infrastructure); OGC; ISO/TC 211; Web map service; WMS; Web coverage service; WCS; GNU; XML; GML; Geo-portal; OGC Web service

Definition

deegree (http://www.deegree.org) is a Java-based open source framework for the creation of spatial data infrastructure (SDI) components. It contains the services needed for SDI (deegree Web Services) as well as portal components (deegree iGeoPortal), mechanisms for handling security issues (deegree iGeoSecurity) and storage/visualization of three-dimensional (3D) geodata (deegree iGeo3D).

deegree is conceptually and interface-wise based on the standards of the Open Geospatial Consortium (OGC) and ISO/TC 211. At the time of writing it is the most comprehensive implementation of OGC standards in one open source framework. The framework is component-based to a high degree, allowing the flexible creation of solutions for a wide variety of use cases.

deegree is the official reference implementation of the OGC for the Web Map Service (WMS) [1] and Web Coverage Service (WCS) [2] standards. It is published under the GNU Lesser General Public License.

Historical Background

deegree is managed in cooperation between the private company lat/lon and the Geographic Information System (GIS) working group of the University of Bonn [3]. The roots of deegree go back to a project of the University of Bonn named EXSE (GIS-Experimental server at the Internet) in the year 1997. The aim of the project was an experiment-based analysis of merging GIS functionality and Internet technology. During the following 3 years several tools and software modules had been developed including a first implementation of the OGC Simple Feature for CORBA specification as an Open Source Java API (Application Programming Interface) (sf4j–Simple Features for Java).

In spring 2001, the sf4j project, the existing tools and software modules were rearranged into a new project called Java Framework for Geospatial Solutions (JaGo) aiming to realize an open source implementation of the OGC web service specifications. The first service implemented was the OGC WMS 1.0.0 specification in summer 2001. By the end of that year WFS (Web Feature Service) 1.0.0 and WCS 0.7 followed. As the important web domains (.de, .org, .net) for JaGo were not available it was decided at the end of 2001 to rename the project "deegree". At that time, deegree had the version number 0.7, the framework contained implementations for OGC WMS 1.0.0, WFS 1.0.0, WCS 0.7 and Web Service Stateless Catalog Profile specifications and a geometry model based on ISO 19107.

The next important step was the release of deegree 1.0 in late summer 2002. Some changes in the architecture offered a better handling of the available OGC Web Services (OWS). An implementation of a multithreading service engine and interfaces to remote OWS enabling high scalability of applications were added. The following minor versions featured new functions like a transactional WFS [4,5], a Gazetteer (WFS-G, [6]) and support of additional data sources. From this time on, deegree WMS supported SLD (Styled Layer Descriptor) [7] and a Catalog Service [8]. Security mechanisms were added to the framework. An important step for increasing the publicity of the project was moving it to sourceforge as its distribution platform. Several developers started reviewing the deegree code base and adding code to the project.

An additional working thread in the development of the framework was started in 2003. It aims at offering components to enable developers to create web clients based on deegree [9]. This new client framework is named iGeoPortal and is part of deegree and supports the OGC standard Web Map Context 1.0.0 [10].

One of the most important steps in deegree development was participation in the OGC Conformance and Interoperability Test and Evaluation Initiative (CITE) project in summer 2003, that resulted in deegree becoming the official OGC reference implementation for WMS 1.1.1 specification. The participation of lat/lon and deegree in CITE was so successful that lat/lon has been charged by the OGC to develop WCS 1.0.0 and WMS 1.3 reference implementations with deegree in the context of OGC's OWS-2 and OWS-4 initiatives.

In 2005, deegree2 was launched, again representing a great step forward in the development of the framework. The keystones of deegree2 are a model-based mechanism for deegree WFS, allowing flexible implementation of different data models using Geography Markup Language (GML) application schemas. Additionally, the development of a portlet-based client framework called deegree iGeoPortal–portlet edition, support for 3D data structures and a web processing service (WPS) [11,12] implementation are included in deegree2.

Scientific Fundamentals

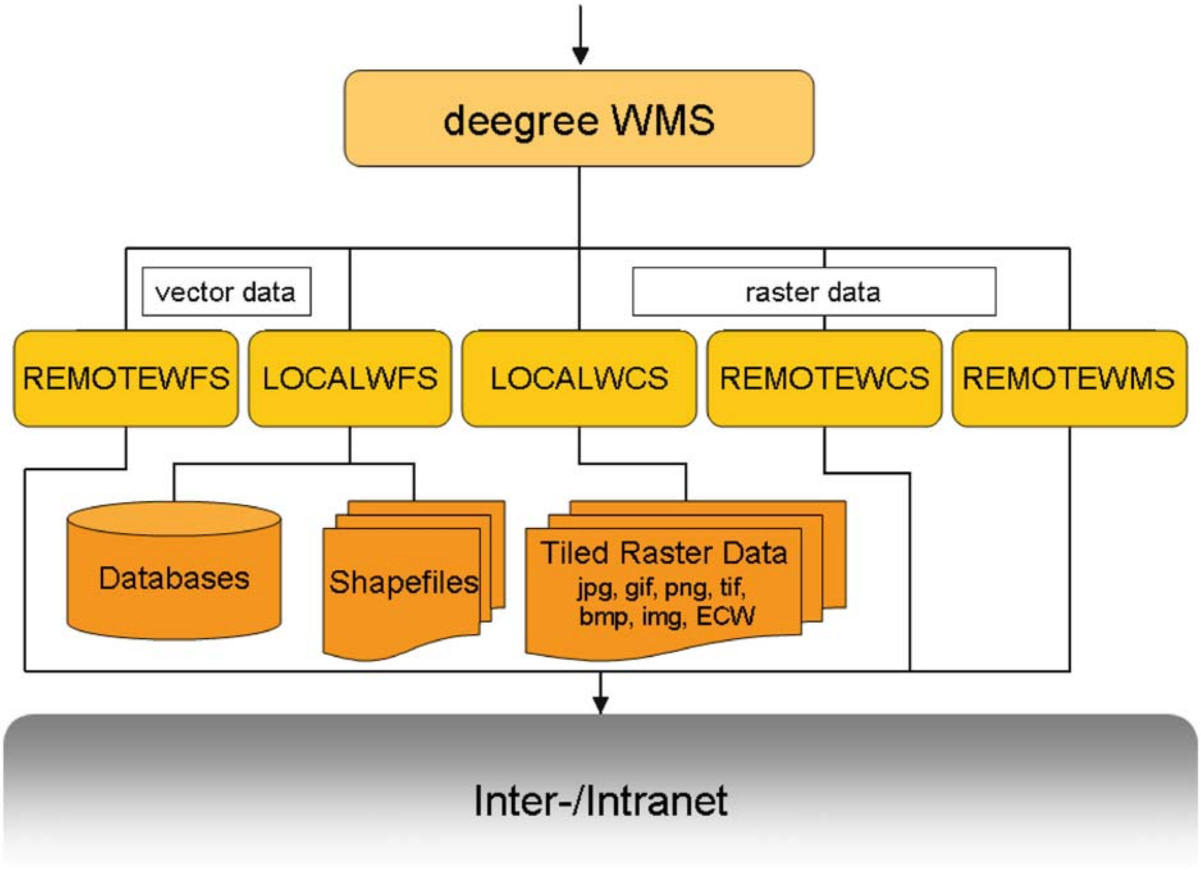
A classical GIS is a monolithic and complex application that requires deep technical knowledge on the user side. Besides a small number of experts who are willing and able to use such systems there exists a large number of users who are just expecting to get a useful and simple answer to a more or less clearly formulated question. To satisfy this need it is not acceptable for most users to buy, install and work with a classical GIS.

Key words of the last years describing the emergence of a new generation of GIS software are "Web-GIS" and "spatial web services". Both are closely related to the specifications of the OGC and the ISO/TC211. These developments highlight a paradigm shift in GIS software design. Instead of having large and very complex monolithic systems a more flexible approach is in the process of establishing itself: moving GIS functionality to Inter- and Intranet Web applications. At the first stage of this development Web GIS was limited to read-only (or view-only) spatial information systems, with an emphasis on maps produced by web map servers. Emerging computer power, increasing public availability of the Internet and the increasing need for more sophisticated spatial information and services lead to the development of additional spatial web services. Related to this, standardization of spatial web services became more important. Today the OGC is widely accepted as a central organization for specifying GIS related services and formats for geodata exchange. So today it is possible to realize complex SDIs using OWS including data visualization [WMS and Web Terrain Service (WTS)], data access (WCS and WFS), data manipulation (WFS-T) and data exploitation (Catalog Service, Gazetteer Service) by connecting different standardized spatial web services through a network. Each of these services can be interpreted as a module that can be connected to one or more other modules through standardized interfaces.

deegree as an open source/free software java project aims to offer these services as well as a client framework and API for more basic GIS functions to enable the realization of highly configurable and flexible SDIs.

The architecture of deegree uses OGC concepts and standards for its internal architecture as well as for its external interfaces. This idea is best described using deegree Web Map Service as an example. Figure 1 shows the different kinds of data sources deegree WMS is able to handle and how they are integrated into the overall architecture.

deegree WMS supports WMS versions 1.1.0, 1.1.1 and 1.3. It can access vector and raster data. For access to vector data, the WFS interface is used inside deegree, while for raster data access, the WCS interface is used. The terms "REMOTEWFS" and "REMOTEWCS" denote the possi-



deegree Free Software, Figure 1 Architecture diagram of deegree Web Map Service (*WMS*). *WCS* Web Coverage Service, WFS

bility of using WFS and WCS services as data sources and displaying their data. In the case that the data access is realized by deegree itself, an internal (faster) interface is used that behaves similar to WFS/WCS but exists as part of the deegree API and can therefore be used in the same Java Virtual Machine (JVM). In this case, the LOCALWFS and LOCALWCS data sources are instantiated. A local WFS can use databases with spatial extension like PostGIS or Oracle, and all kinds of other relational databases using a concept called GenericSQLDataStore or Shapefiles. A local WCS can use file-based rasterdata or Oracle GeoRaster. The last possibility is to access remote WMS Services ("REMOTEWMS"), cascading their maps inside deegree.

The concept of reusing OGC and ISO standards and interfaces inside the architecture is a special characteristic of deegree.

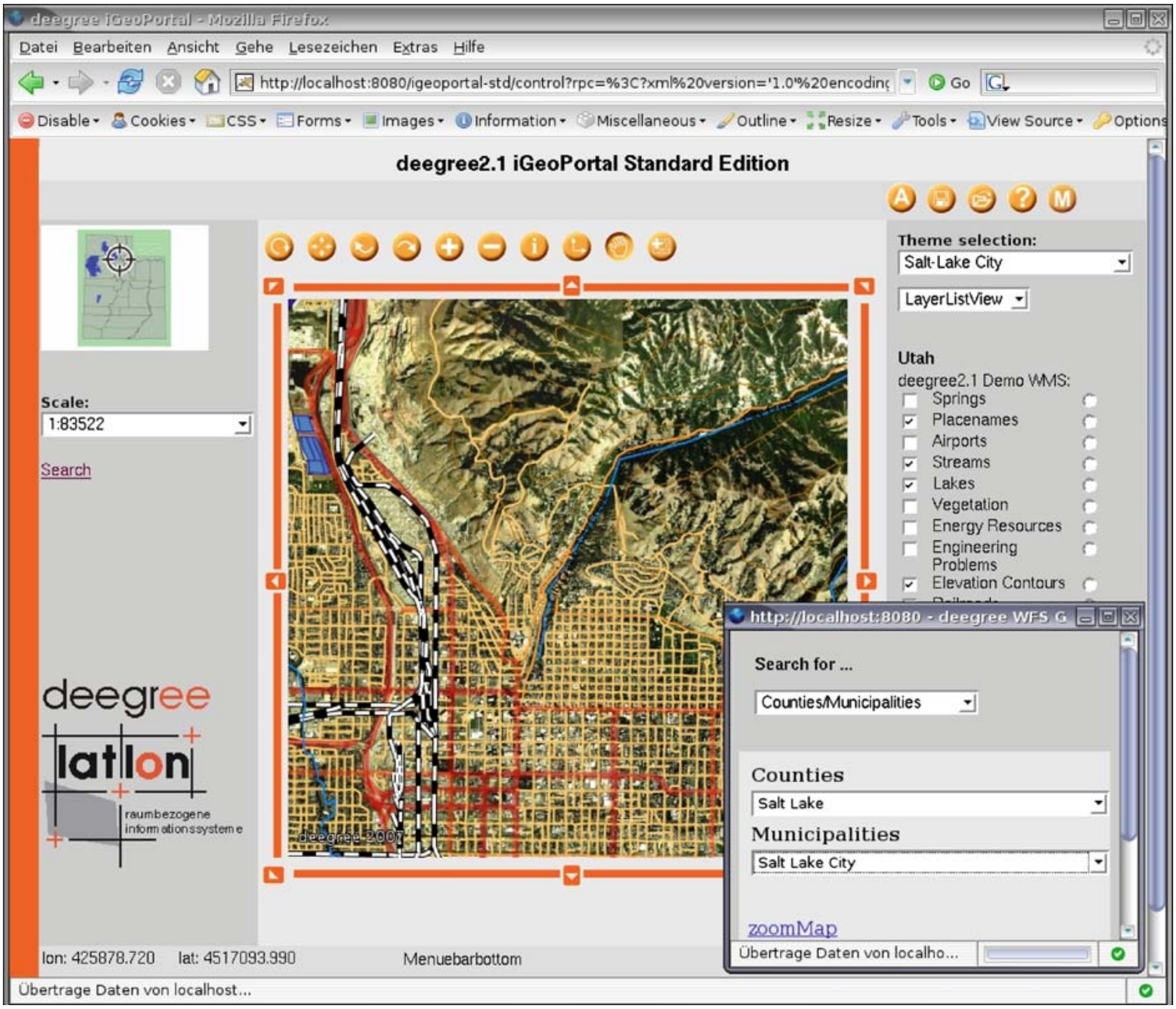
All configuration files of deegree are **Extensible Markup Language** (XML)-based and reuse relevant OGC specifications wherever possible. The configuration documents for deegree WFS for example consist of extended GML application schemas and WFS capabilities files. This mechanism makes configuring deegree easier for people familiar with OGC specifications, while at the same time making working with deegree a practical OGC tutorial.

Key Applications

deegree is mainly used in systems where interoperability is an important issue. This includes the following application areas.

Geoportals

Different kinds of SDI portals are created using deegree iGeoPortal standard and portlet edition and the corresponding geo-webservices. While deegree standard edition is using DHTML (Dynamic HTML) technology, the portlet edition is based on the JSR-168, the portlet standard. Both editions use AJAX technology for some specific modules. These portals include standard Web-GIS and specialized applications as used by municipalities, environmental, surveying and other agencies.



deegree Free Software, Figure 2 deegree iGeoPortal

deegree iGeoPortal itself consists of different modules for separate, but combinable, functionalities. The list of modules includes: map, download, gazetteer, catalog and security (Fig. 2).

3D Data Storage and Visualization

deegree can be used to store 3D geodata such as digital terrain and building models in file-based systems and relational databases. Using different deegree web services, this data can be queried and displayed. These systems are often used by surveying agencies and municipalities to store their 3D data for commercial purposes or to fulfil EU regulations such as the European noise guidelines. deegree iGeo3D uses the OGC standards WFS, WCS, WTS and CityGML.

Metadata and Catalogs

Catalog services are key components for SDIs. Metadata based on the ISO standards 19115 (metadata for geodata) and 19119 (metadata for geoservices) and 19139 (XML encoding of 19115) can be used to describe georesources in a standardized way. deegree includes functionalities for the creation, storage, querying, retrieval and display of metadata. These systems are used by all kinds of institutions in need of handling large amounts of geodata. deegree implements the core functionality of catalog services [8] as well as the so-called ISO Application Profile [13].

Security Components

Access control for geodata and services is an important issue for a wide variety of applications. deegree iGeoSecurity can be used to define access mechanisms using authentication and authorization mechanisms, secure connections and filtering of geodata. A database for managing users, user groups, roles and rights called deegree U3R is the core of the security components.

Desktop GIS

An additional application area of deegree is the classical desktop GIS. Using deegree components, the open source desktop GIS Java Unified Mapping Platform (JUMP) was enhanced to support WMS and WFS. A number of additional modifications and extensions was also developed.

Future Directions

The deegree framework is continually enhanced and extended. Three main areas will be focal points for the future of deegree. These are:

- Professional geoportals using technologies such as JSR-168, AJAX and web service standards like WSDL. Key challenges here will be modularity and usability while maintaining good performance.
- 3D solutions for efficient and extensible handling of digital 3D data. Fast access to large amounts of data and flexible storage and visualization of different levels of detail of building models are the major tasks. The establishment of CityGML as OGC standard in the future is an important aspect for this.
- Metadata storage and creation using international standards. Standardization and workflow modeling is an important aspect in this application area. Of major importance is support of the publish-bind-find paradigm, that allows dynamic binding of OWS during runtime of a system.

Cross References

► Data Infrastructure, Spatial
► Geography Markup Language (GML)
► Information Services, Geography
► Metadata and Interoperability, Geospatial
► National Spatial Data Infrastructure
► OGC's Open Standards for Geospatial Interoperability
► Web Mapping and Web Cartography
► Web Services, Geospatial

References

1. de La Beaujardière J (ed) (2003) OpenGIS Web Map Server Interface Implementation Specification Revision 1.1.1. OpenGIS Project Document 01-068r3 2003. http://portal.opengeospatial.org/files/?artifact_id=1081&version=1&format=pdf
2. Evans J (ed) (2002) Web Coverage Service (WCS) Version 1.0.0. OpenGIS Project Document 03-065r6. https://portal.opengeospatial.org/files/?artifact_id=3837
3. Fitzke, J., Greve, K., Müller, M., Poth, A.. Building SDIs with Free Software – the deegree project. In: Proceedings of CORP 2004, Vienna, Austria, 25–17 February, 2004. Available via http://www.corp.at
4. Vretanos PA (ed) OpenGIS Web Feature Service Implementation Specification Version 1.1.0. OpenGIS Project Document 04-094. https://portal.opengeospatial.org/files/?artifact_id=8339
5. Vretanos PA (ed) (2002) OpenGIS Filter Encoding Specification Version 1.1.0. OpenGIS Project Document 04-095. http://portal.opengeospatial.org/files/?artifact_id=8340
6. Atkinson R, Fitzke J (ed) (2006) Gazetteer Service Profile of the Web Feature Service Implementation Specification. OpenGIS Project Document 05-035r2. http://portal.opengeospatial.org/files/?artifact_id=15529
7. Lalonde W (ed) (2002) Styled Layer Descriptor Implementation Specification. OpenGIS Project Document 02-070. https://portal.opengeospatial.org/files/?artifact_id=1188
8. Nebert D (ed) (2002) OpenGIS Catalog Service Specification. OpenGIS Project Document 04-021r3. http://portal.opengeospatial.org/files/?artifact_id=5929&version=2
9. Müller, M., Poth, A.: deegree–components and framework for creation of SDIs. Geoinformatics **7**, 2004
10. Humblet JP (ed) (2003) Web Map Context Documents. OpenGIS Project Document 03-036r2. https://portal.opengeospatial.org/files/?artifact_id=3841
11. Kiehle, C., Greve, K. Standardized Geoprocessing – Taking Spatial Data Infrastructures One Step Further. In: Proceedings of AGILE, Minneapolis, Minnesota, USA, 23–28 July 2006
12. Schut, P., Whiteside, A. (ed.) OpenGIS Web Processing Service. OGC project document 05-007r4. OGC (2005)
13. Voges U, Senkler K (ed) (2004) OpenGIS Catalogue Services Specification 2.0 – ISO19115/ISO19119 Application Profile for CSW 2.0 OpenGIS Project Document 04-038r2. https://portal.opengeospatial.org/files/?artifact_id=8305

Deegree Library

► deegree Free Software

Deegree Open Source Framework

► deegree Free Software

Delaunay Triangulation

► Constraint Databases and Data Interpolation

DEM

► Oracle Spatial, Raster Data
► Photogrammetric Products

Dempster Shafer Belief Theory

► Computing Fitness of Use of Geospatial Datasets

Dependence, Spatial

► Spatial Regression Models

Detection of Changes

► Change Detection

Determinant

► Hurricane Wind Fields, Multivariate Modeling

DGC

► Distributed Geospatial Computing (DGC)

Digital Change Detection Methods

► Change Detection

Digital Divide

► Data Infrastructure, Spatial

Digital Earth

► Multimedia Atlas Information Systems

Digital Elevation Model

► Photogrammetric Products

Digital Image

► Raster Data

Digital Image Processing

► Evolution of Earth Observation

Digital Line Graph

► Spatial Data Transfer Standard (SDTS)

Digital Mapping

► Rules-Based Processing in GIS

Digital Road Networks

► Road Maps, Digital

Digital Surface Model

► Photogrammetric Products

Digitization of Maps

► Positional Accuracy Improvement (PAI)

Dijktra's Shortest Path Algorithm

► Fastest-Path Computation

Dimension Reduction

► Hurricane Wind Fields, Multivariate Modeling

Dimensionally Extended Nine-Intersection Model (DE-9IM)

CHRISTIAN STROBL
German Remote Sensing Data Center (DFD),
German Aerospace Center (DLR), Weßling, Germany

Synonyms

Nine-intersection model; 9IM; DE-9IM; Four-intersection model; 4IM; Egenhofer operators; Clementini operators; Topological operators

Definition

The Dimensionally Extended Nine-Intersection Model (DE-9IM) or Clementini-Matrix is specified by the OGC "Simple Features for SQL" specification for computing the spatial relationships between geometries. It is based on the Nine-Intersection Model (9IM) or Egenhofer-Matrix which in turn is an extension of the Four-Intersection Model (4IM).

The Dimensionally Extended Nine-Intersection Model considers the two objects' interiors, boundaries and exteriors and analyzes the intersections of these nine objects parts for their relationships (maximum dimension (−1, 0, 1, or 2) of the intersection geometries with a numeric value of –1 corresponding to no intersection).

The spatial relationships described by the DE-9IM are "Equals", "Disjoint", "Intersects", "Touches", "Crosses", "Within", "Contains" and "Overlaps".

Historical Background

Regarding the description of topological relationships of geodata, three common and accepted approaches exist. Each of these systems describes the relationship between two objects based on an intersection matrix.

- Four-Intersection Model (4IM): Boolean set of operations (considering intersections between boundary and interior)
- Nine-Intersection Model (9IM): Egenhofer operators (taking into account exterior, interior and boundary of objects)
- **Dimensionally Extended Nine-Intersection Model (DE-9IM):** Clementini operators using the same topological primitives as Egenhofer, but taking the dimension type of the intersection into consideration.

The three intersection models are based on each other. The Dimensionally Extended Nine-Intersection Model [3,4,5] dimensionally extends the Nine-Intersection Model (9IM) of Egenhofer and Herring [9]. The Nine-Intersection Model in turn extends the Four-Intersection Model (4IM) from Egenhofer [7,9,12] by adding the intersections with the exteriors [10,11].

The **Dimensionally Extended Nine-Intersection Model (DE-9IM)** is accepted by the ISO/TC 211 [14] and by the Open Geospatial Consortium [15], and will be described in the following paragraphs.

Scientific Fundamentals

Each of the mentioned intersection models is based on the accepted definitions of the boundaries, interiors and exteriors for the basic geometry types which are considered. Therefore, the first step is defining the interior, boundary and exterior of the involved geometry types. The domain considered consists of geometric objects that are topologically closed.

- Boundary: The boundary of a geometry object is a set of geometries of the next lower dimension.
- The interior of a geometry object consists of those points that are left (inside) when the boundary points are removed.
- The exterior of a geometry object consists of points not in the interior or boundary.

The next step is to consider the topological relationship of two geometry objects. Each geometry is represented by its Interior (I), Boundary (B) and Exterior (E), thus all possible relationships of two geometry objects can be described by a 3×3-matrix. If the values of the matrix are the dimension of the respective relationship of the two geometry objects, e. g., between the interior of geometry object A and the boundary of geometry object B, the result is the Dimensionally Extended Nine-Intersection Matrix (DE-9IM) after Clementini [5]. This matrix has the form

$$DE-9IM(A,B) = \begin{bmatrix} \dim(I(A)\cap I(B)) & \dim(I(A)\cap B(B)) & \dim(I(A)\cap E(B)) \\ \dim(B(A)\cap I(B)) & \dim(B(A)\cap B(B)) & \dim(B(A)\cap E(B)) \\ \dim(E(A)\cap I(B)) & \dim(E(A)\cap B(B)) & \dim(E(A)\cap E(B)) \end{bmatrix}.$$

Topological predicates are Boolean functions that are used to test the spatial relationships between two geometry objects. The Dimensionally Extended Nine-Intersection Model provides eight such spatial relationships between points, lines and polygons (q.v. [15] and Table 2).

The following describes each topological predicate by example:

"Equals": Example DE-9IM for the case where A is a Polygon which is equal to a Polygon B.

"Disjoint": Example DE-9IM for the case where A is a Line which is disjoint to a MultiPoint object B. NB: The boundary of a Point is per definition empty (−1).

"Intersects": Example DE-9IM for the case where A is a Line which intersects a Line B. NB: The "Intersects"-relationship is the inverse of Disjoint. The Geometry objects have at least one point in common, so the "Intersects" relationship includes all other topological predicates. The example in Fig. 3 is therefore also an example for a "Crosses"-relationship.

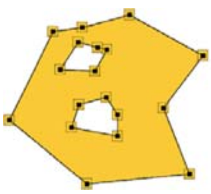
"Touches": Example DE-9IM for the case where A is a Polygon that touches two other Polygons B and C. The DE-9IM for both relationships differs only in the dimension of the boundary-boundary-intersection which has the value 1 for the relationship A/B and the value 0 for the relationship A/C.

"Crosses": Example DE-9IM for the case where A is a Polygon and B is a Line that crosses line A.

"Overlaps": Example DE-9IM for the case where A is a Line which overlaps the Line B. The overlaps-relationship is not commutative. Line A overlaps Line B is different from Line B overlaps Line A. The consequence of this not-commutative relationship is that the DE-9IM differs yet in the interior-boundary-respectively in the boundary-interior-relationship (bold printed).

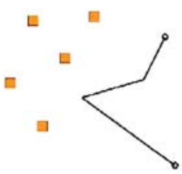
Dimensionally Extended Nine-Intersection Model (DE-9IM), Table 1 Definition of the Interior, Boundary and Exterior for the main geometry types which are described by the Open Geospatial Consortium [15]

Geometric Subtypes	Dim	Interior (I)	Boundary (B)	Exterior (E)
Point, MultiPoint	0	Point, Points	Empty set	Points not in the interior or boundary
LineString, Line	1	Points that are left when the boundary points are removed	Two end Points	Points not in the interior or boundary
LinearRing	1	All Points along the LinearRing	Empty set	Points not in the interior or boundary
MultiLineString	1	Points that are left when the boundary points are removed	Those Points that are in the boundaries of an odd number of its element Curves	Points not in the interior or boundary
Polygon	2	Points within the Rings	Set of Rings	Points not in the interior or boundary
MultiPolygon	2	Points within the Rings	Set of Rings of its Polygons	Points not in the interior or boundary



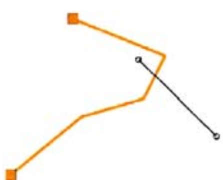
	Interior (B)	Boundary (B)	Exterior (B)
Interior(A)	2	-1	-1
Boundary (A)	-1	1	-1
Exterior (A)	-1	-1	2

Dimensionally Extended Nine-Intersection Model (DE-9IM), Figure 1 Example for an "Equals"-relationship between a Polygon A and a Polygon B



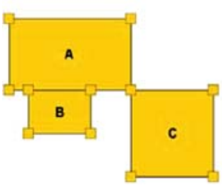
	Interior (B)	Boundary (B)	Exterior (B)
Interior(A)	-1	-1	1
Boundary (A)	-1	-1	0
Exterior (A)	0	-1	2

Dimensionally Extended Nine-Intersection Model (DE-9IM), Figure 2 Example for a "Disjoint"-relationship between a Line A and MultiPoint B



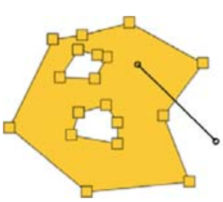
	Interior (B)	Boundary (B)	Exterior (B)
Interior(A)	0	-1	1
Boundary (A)	-1	-1	0
Exterior (A)	1	0	2

Dimensionally Extended Nine-Intersection Model (DE-9IM), Figure 3 Example for a "Disjoint"-relationship between a Line A and a MultiPoint B



	Interior (B)	Boundary (B)	Exterior (B)
Interior(A)	-1	-1	2
Boundary (A)	-1	**1/0**	1
Exterior (A)	2	1	2

Dimensionally Extended Nine-Intersection Model (DE-9IM), Figure 4 Example for a "Touches"-relationship between three Polygons A, B and C

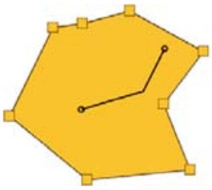


	Interior (B)	Boundary (B)	Exterior (B)
Interior(A)	1	0	2
Boundary (A)	0	-1	1
Exterior (A)	1	0	2

Dimensionally Extended Nine-Intersection Model (DE-9IM), Figure 5 Example for a "Crosses"-relationship between a Polygon A and a Line B

	Interior (B)	Boundary (B)	Exterior (B)
Interior(A)	1	**-1/0**	1
Boundary (A)	**0/-1**	-1	0
Exterior (A)	1	0	2

Dimensionally Extended Nine-Intersection Model (DE-9IM), Figure 6 Example for an "Overlaps"-relationship between two Lines A and B



	Interior (B)	Boundary (B)	Exterior (B)
Interior(A)	1	-1	-1
Boundary (A)	0	-1	-1
Exterior (A)	2	1	2

Dimensionally Extended Nine-Intersection Model (DE-9IM), Figure 7 Example for a "Within"-relationship between a Line A and a Polygon B

	Interior (B)	Boundary (B)	Exterior (B)
Interior(A)	0	-1	0
Boundary (A)	-1	-1	-1
Exterior (A)	-1	-1	2

Dimensionally Extended Nine-Intersection Model (DE-9IM), Figure 8 Example for a "Contains"-relationship between two MultiPoints A and B

Dimensionally Extended Nine-Intersection Model (DE-9IM), Table 2 Topological predicates and their corresponding meanings after the Dimensionally Extended Nine-Intersection Model, from [6]

Topological Predicate	Meaning
Equals	The Geometries are topologically equal
Disjoint	The Geometries have no point in common
Intersects	The Geometries have at least one point in common (the inverse of Disjoint)
Touches	The Geometries have at least one boundary point in common, but no interior points
Crosses	The Geometries share some but not all interior points, and the dimension of the intersection is less than that of at least one of the Geometries
Overlaps	The Geometries share some but not all points in common, and the intersection has the same dimension as the Geometries themselves
Within	Geometry A lies in the interior of Geometry B
Contains	Geometry B lies in the interior of Geometry A (the inverse of Within)

"Within": Example DE-9IM for the case where A is a Line which lies within the Polygon B.
"Contains": Example DE-9IM for the case where A is a MultiPoint Object (squares) which contains another MultiPoint B (circles).
The pattern matrix represents the DE-9IM set of all acceptable values for a topological predicate of two geometries. The pattern matrix consists of a set of 9 pattern-values, one for each cell in the matrix. The possible pattern values p are (T, F, *, 0, 1, 2) and their meanings for any cell where x is the intersection set for the cell are as follows:

$$
\begin{aligned}
p &= \mathrm{T} &&\Rightarrow \dim(x) \in (0, 1, 2)\,, \quad \text{i.e. } x \neq \emptyset\\
p &= \mathrm{F} &&\Rightarrow \dim(x) = -1\,, \quad \text{i.e. } x = \emptyset\\
p &= * &&\Rightarrow \dim(x) \in (-1, 0, 1, 2)\,, \quad \text{i.e., Don't Care}\\
p &= 0 &&\Rightarrow \dim(x) = 0\\
p &= 1 &&\Rightarrow \dim(x) = 1\\
p &= 2 &&\Rightarrow \dim(x) = 2\,.
\end{aligned}
$$

The pattern matrices for the eight topological predicates of the DE-9IM are described in Table 3.
One additional topological predicate is the Relate predicate based on the pattern matrix. The Relate predicate has the advantage that clients can test for a large number of spatial relationships which topological predicate is the appropriate one. With the Relate method defined by [15], the pattern matrix after the DE-9IM can be determined, e. g., in PostGIS

```
SELECT RELATE(a.geom,b.geom)
  FROM country a, river b
    WHERE a.country_name='Bavaria'
    AND b.river_name='Isar';
-------
1020F1102
```

The comparison with the pattern matrices from Table 3 shows the "Crosses"-predicate as a result for the topological relationship between the country "Bavaria" and the river "Isar".

Key Applications

The Dimensionally Extended Nine-Intersection Model is mainly used in the field of spatial databases like PostGIS, Oracle Spatial, ArcSDE or Spatial Support for DB2 for z/OS (formerly known as DB2 Spatial Extender). Additionally, the DE-9IM is also integrated into GIS libraries like JTS and GEOS, and desktop GIS like Jump and GeOxygene.

Future Directions

A lot of work which extends the DE-9IM has already been done. Extensions exist which consider regions with holes [11], composite regions [2] and heterogeneous geometry-collection features [16]. The pitfall is in the lack of integration within this work regarding the common GIS-software mentioned above.
Further directions will probably involve the use of the DE-9IM in the broad field of the geospatial semantic web,

Topological Predicate	Pattern Matrix
A.Equals(B)	$\begin{bmatrix} T & * & F \\ * & * & F \\ F & F & * \end{bmatrix}$
A.Disjoint(B)	$\begin{bmatrix} F & F & * \\ F & F & * \\ * & * & * \end{bmatrix}$
A.Intersects(B)	$\begin{bmatrix} T & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$ **or** $\begin{bmatrix} * & T & * \\ * & * & * \\ * & * & * \end{bmatrix}$ **or** $\begin{bmatrix} * & * & * \\ T & * & * \\ * & * & * \end{bmatrix}$ **or** $\begin{bmatrix} * & * & * \\ * & T & * \\ * & * & * \end{bmatrix}$
A.Touches(B)	$\begin{bmatrix} F & T & * \\ * & * & * \\ * & * & * \end{bmatrix}$ **or** $\begin{bmatrix} F & * & * \\ * & T & * \\ * & * & * \end{bmatrix}$ **or** $\begin{bmatrix} F & * & * \\ T & * & * \\ * & * & * \end{bmatrix}$
A.Crosses(B)	$\begin{bmatrix} T & * & T \\ * & * & * \\ * & * & * \end{bmatrix}$ **or** $\begin{bmatrix} 0 & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$
A.Overlaps(B)	$\begin{bmatrix} T & * & T \\ * & * & * \\ T & * & * \end{bmatrix}$ **or** $\begin{bmatrix} 1 & * & T \\ * & * & * \\ T & * & * \end{bmatrix}$
A.Within(B)	$\begin{bmatrix} T & * & F \\ * & * & F \\ * & * & * \end{bmatrix}$
A.Contains(B)	$\begin{bmatrix} T & * & * \\ * & * & * \\ F & F & * \end{bmatrix}$

Dimensionally Extended Nine-Intersection Model (DE-9IM), Table 3 Topological predicates and the corresponding pattern matrices after the Dimensionally Extended Nine-Intersection Model (DE-9IM), [15]

e. g., [8], development of topological relationships for 3D Objects, e. g., [1], and the extension of the DE-9IM for spatio-temporal databases [13].

Cross References

- ► Mathematical Foundations of GIS
- ► OGC's Open Standards for Geospatial Interoperability
- ► Open-Source GIS Libraries
- ► Oracle Spatial, Geometries
- ► PostGIS
- ► Spatio-temporal Query Languages

Recommended Reading

1. Borrmann, A., van Treeck, C., Rank, E.: Towards a 3D Spatial Query Language for Building Information Models. In: Proceedings of the 11th Int. Conf. on Computing in Civil and Building Engineering (ICCCBE-XI), Montreal, Canada (2006)
2. Clementini, E., Di Felice, P., Califano, G.: Composite Regions in Topological Queries. Information Systems **20**, 579–594 (1995)
3. Clementini, E., Di Felice, P., van Oosterom, P.: A Small Set of Formal Topological Relationships Suitable for End-User Interaction. In: Proceedings of the 3rd Intl. Symposium on Large Spatial Databases, pp. 277–295. Singapore (1993)
4. Clementini, E.,Di Felice, P.A.: Comparison of Methods for Representing Topological Relationships. Information Sciences **80**, 1–34 (1994)
5. Clementini, E., Di Felice, P.A.: Model for Representing Topological Relationships Between Complex Geometric Features in Spatial Databases. Information Sciences **90**(1–4), 121–136 (1996)
6. Davis, M., Aquino, J.: JTS Topology Suite – Technical Specifications. Vivid Solutions Victoria, British Columbia (2003)
7. Egenhofer, M.: A Formal Definition of Binary Topological Relationships. In: Litwin, W., Schek, H.J. (eds.) Proc. 3rd Intl. Conf. on Foundations of Data Organization and Algorithms (FODO). Lecture Notes in Computer Science 367. pp. 457–472. Springer Verlag, Paris (1989)
8. Egenhofer, M.: Toward the Semantic Geospatial Web. Voisard, A., Chen, S.C. (eds.) ACM-GIS 2002. pp. 1–4. McLean, Virginia, USA (2002)
9. Egenhofer, M., Herring, J.: Categorizing Binary Topological Relationships between Regions, Lines, and Points in Geographic Databases. Dept. of Surveying Engineering, Technical Report. University of Maine, Orono, Maine (1991)
10. Egenhofer, M., Sharma, J., Mark, D.: A Critical Comparison of the 4-Intersection and 9-Intersection Models for Spatial Relations: Formal Analysis. In: McMaster, R., Armstrong M. (eds.) Proceedings of AutoCarto 11 Minneapolis (1993)
11. Egenhofer, M.J., Clementini, E., di Felice, P.A.: Topological relations between regions with holes. International Journal of Geographical Information Systems **8**(2), 129–142 (1994)
12. Egenhofer, M.J., Herring, J.: A mathematical framework for the definition of topological relationships. In: Proceedings of the Fourth International Symposium on Spatial Data Handling, pp. 803–813. Zürich, Switzerland (1990)
13. Güting, R.H., Schneider, M.: Moving Objects Databases. Morgan Kaufmann, San Francisco (2005)
14. ISO/TC211 (ed.) (2003): ISO 19107: Geographic information – Spatial schema.
15. OGC (ed.) (2005): OpenGIS® Implementation Specification for Geographic Information – Part 1: Common architecture (Version 1.1.0)

16. Zhong, Z.-N., Jing, N., Chen, L., Wu, Q.-Y.: Representing topological relationships among heterogeneous geometry-collection features. Journal of Computer Science and Technology archive, Institute of Computing Technology, Beijing, China **19**(3), 280–289 (2004)

Directed Acyclic Graphs

► Bayesian Network Integration with GIS

Directory Rectangles

► R*-tree

Dirichlet Tessellation

► Voronoi Diagram

Discord or Non-Specificity in Spatial Data

► Uncertainty, Semantic

Discretization of Quatitative Attributes

► Geosensor Networks, Qualitative Monitoring of Dynamic Fields

Disease Mapping

► Public Health and Spatial Modeling

Disk Page

► Indexing Schemes for Multi-dimensional Moving Objects

Distance Measures

► Indexing and Mining Time Series Data

Distance Metrics

JAMES M. KANG
Department of Computer Science and Engineering,
University of Minnesota, Minneapolis, MN, USA

Synonyms

Euclidean distance; Manhattan distance

Definition

The Euclidean distance is the direct measure between two points in some spatial space. These points can be represented in any n-dimensional space. Formally, the Euclidean distance can be mathematically expressed as:

$$\sqrt{(a_1 - b_1)^2 + (a_2 - b_3)^2 + \cdots + (a_n - b_n)} \qquad (1)$$

where a and b are two points in some spatial space and n is the dimension.
The Manhattan distance can be mathematically described as:

$$|x_1 - x2| + |y1 - y2| \qquad (2)$$

where A and B are the following points (x_1, y_1) and (x_2, y_2), respectively. Notice that it does not matter which order the difference is taken from because of the absolute value condition.

Main Text

The Euclidean distance can be measured at a various number of dimensions. For dimensions above three, other feature sets corresponding to each point could be added as more dimensions within a data set. Thus, there can be an infinite number of dimensions used for the Euclidean distance. For a Voronoi Diagram in a two dimensional space, a distance metric that can be used is the Euclidean distance where the number of dimensions n would be two.
A common distance metric that uses the Euclidean distance is the Manhattan distance. This measure is similar to finding the exact distance by car from one corner to another corner in a city. Just using the Euclidean distance could not be used since we are trying to find the distance where a person can physically drive from the starting to the ending point. However, if we measure the distances between intersections using the Euclidean distance and add these values together, this would be the Manhattan distance.

Cross References

► Voronoi Diagram

► Voronoi Diagrams for Query Processing
► Voronoi Terminology

Distance-Preserving Mapping

► Space-Filling Curves

Distributed Algorithm

► Geosensor Netwoks, Estimating Continuous Phenonena

Distributed Caching

► Olap Results, Distributed Caching

Distributed Computing

► Distributed Geospatial Computing (DGC)
► Grid, Geospatial

Distributed Databases

► Smallworld Software Suite

Distributed Geocomputation

► Distributed Geospatial Computing (DGC)

Distributed Geospatial Computing (DGC)

CHAOWEI PHIL YANG
Joint Center for Intelligent Spatial Computing (CISC), College of Science, George Mason University, Fairfax, VA, USA

Synonyms

DGC; Distributed geospatial information processing; Distributed geocomputation; Distributed computing; Parallel computing

Definition

Distributed geospatial computing (DGC) refers to the geospatial computing that resides on multiple computers connected through computer networks. Figure 1 illustrates DGC within the client/server (C/S) architecture [1]: where the geospatial computing is conducted by the geospatial components, which can communicate with each other or communicate through wrapping applications, such as web server and web browser. The geospatial components can communicate through application level protocols, such as the hypertext transfer protocol (HTTP) or other customized protocols.

Geospatial computing includes utilizing computing devices to collect, access, input and edit, archive, analyze, render/visualize geospatial data, display within user interface, and interact with end users. Figure 2 illustrates that these previously tightly coupled components are now decoupled and distributed to a number of computers as either servers or clients across a computer network. The communications among these components are supported through different protocols: for example, the data exchange can be structured querying language (SQL) [2], the geospatial protocol can be arc extensible markup language (ArcXML) [3], the message can be transmitted through pipe, and the client to web service can be HTTP [4].

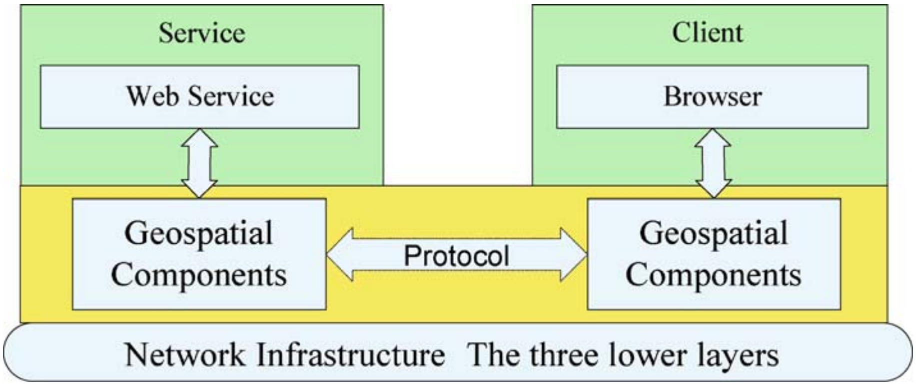
This distribution of geospatial computing components matches the needs to integrate the legacy and future components of geospatial computing deployed at different computers and hosted by different organizations [5].

Because the intensive computing component mainly resides in the geospatial analysis component, geospatial computing is focused on this component.

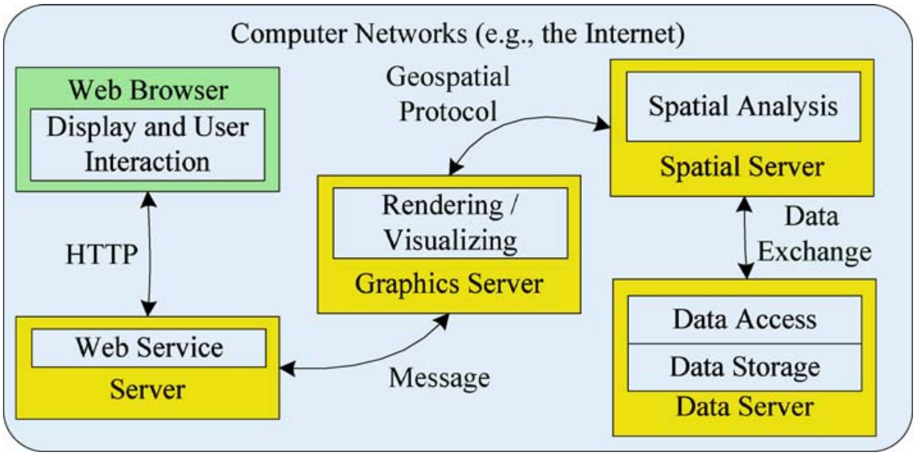
Historical Background

Figure 3 illustrates the historical evolution of DGC. The major development of DGC can be traced back to the beginning of computer networks when the Defense Advanced Research Projects Agency (DARPA) processed geospatial information across their intranet. Xerox's mapping server is recognized as the first system for processing distributed spatial information across the internet [6]. In 1994, the Federal Geographic Data Committee (FGDC) [7] was established to share geospatial computing across distributed platform, and the Open Geospatial Consortium (OGC) [8] and the International Standards Organization/Technical Committee 211 (ISO/TC211) [9] were established to define a set of standardized interfaces to share the DGC platform.

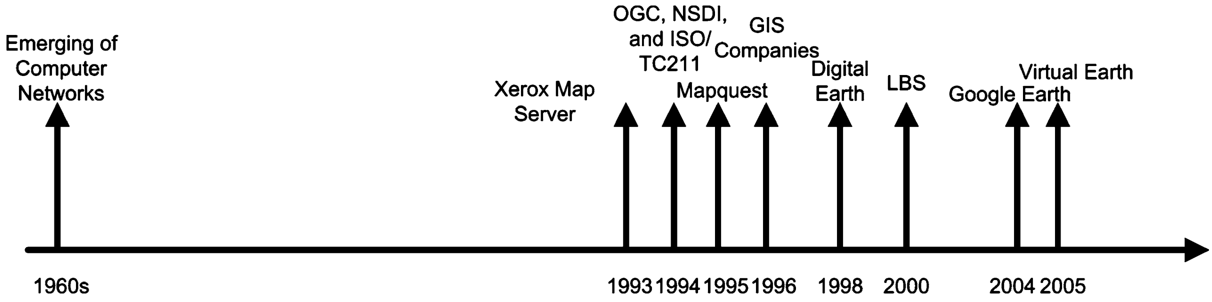
In 1995, Mapquest and other DGC applications were released and eventually achieved great success by leveraging a single geospatial computing use, such as routing, to serve the public. In 1996, the Environmental System Research Institute (ESRI), Intergraph, and other geographic information systems (GIS) companies began to participate in the DGC effort by fully implementing geospa-



Distributed Geospatial Computing (DGC), Figure 1 Distributed geospatial computing (DGC) within computer network architecture



Distributed Geospatial Computing (DGC), Figure 2 Distributions of DGC components onto servers within a computer network

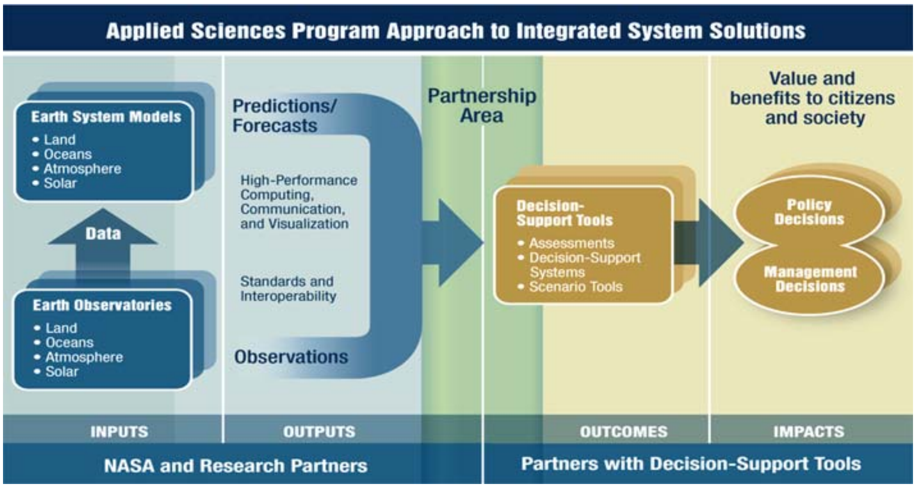


Distributed Geospatial Computing (DGC), Figure 3 Evolution of DGC. *OGC* The Open Geospatial Consortium, *ISO/TC211* International Standards Organization/Technical Committee 211, *NSDI* National Saptial Data Infrastructure, *GIS* Geographic/Geospatial Information System, *LBS* Location-based Services

tial computing components in the distributed environment [6,10].

In 1998, Vice President Al Gore proposed the Digital Earth vision to integrate all geospatial resources to support a virtual environment that could facilitate all walks of human life from research and development, to daily life. In 2004, Google Earth was announced, and provided a milestone to gradually implement such a vision. In 2005, Microsoft started Virtual Earth. Within these two implementations, limited functions of geospatial computing are addressed, but they focus on the massive data and friendly user interaction, and solved many problems in dealing with massive simultaneous users by using thousands to millions of computers [11].

Accompanying these events, which have profoundly impacted on DGC, many scholars also addressed issues on how to distribute geospatial computing [12], improve performance of DGC [13], parallelize geospatial computing [14], and leverage grid platforms and agent-based environments [15] for distributed geospatial computing. Sever-



Distributed Geospatial Computing (DGC), Figure 4 DGC is embedded in the integrated system solutions [21]

al centers/labs, such as the Joint Center for Intelligent Spatial Computing (CISC) [16] and Pervasive Technology Lab (PTL) [17] are established to address the research needs in this field.

Scientific Fundamentals

The scientific fundamentals of DGC rely on geometry, topology, statistics, and cartography principles to design and implement algorithms. Geospatial recognition and physical science provides conceptualization and model foundations for DGC. Earth sciences provide support for the DGC application design. Other relevant sciences, such as human recognition, provide support for other DGC aspects, such as graphical user interface.

The architecture of DGC also relies on the combinational scientific research, and is mainly driven by the distributed computing advancements, for example, the C/S, three-tier architecture, *N*-tier architecture, tightly coupled, and peer-to-peer categories.

DGC is heavily dependent on the concurrency process, with processing tasks increasing in earth sciences and emergency or rapid response systems. Therefore, multiprocessor systems, multicore systems, multicomputer systems, and computer clusters, as well as grid computing are being researched to provide support to DGC.

Key Applications

DGC are used in many application domains, most notably the sciences and domains needing processing of distributed geospatial resources, such as oceanography.

Sciences

The first domain to use DGC is the sciences. The science domains discussed here are geography, oceanography, geology, health.

Geography DGC can be used to help integrate widely geographically dispersed geospatial resources and provide a comprehensive overview of the earth surface [18].

Oceanography DGC can be used to help integrate the in-situ and satellite observation system and the modeling system to monitor tsunami, sea level changes, and coastal disasters [19].

Geology DGC can help to integrate the observed earth surface heat flux and the in-situ sensor's observation to monitor and possibly predict earth quakes [20].

Health DGC can help to integrate the health information and environment observations to find correlation between environmental changes and human health.

National and GEOSS Applications

NASA identified 12 application areas of interest at the national level [21] and GEO identified 9 application areas of interest at the global level [22]. Figure 4 illustrates the integration of earth observations, earth system models, with decision support tools to support decision or policy making [21]. All these areas require the use of DGC to integrate distributed earth observations, earth system models, and to support decision-support tools hosted by government agencies or other organizations. The 12 application areas are agricultural efficiency, air quality, aviation safety, carbon management, coastal management, disaster management, ecological forecasting, energy management, homeland security, invasive species, public health, and water management. The 9 application areas are human health and well-being, natural and human-induced disasters, energy resources, weather information and forecasting, water resources, climate variability and change, sustainable agriculture and desertification, ecosystems, and oceans.

Routing

DGC can be used to integrate road network datasets, dispatching a routing request to different servers to select the shortest or fastest path. This can be used in (1) driving directions, such as Mapquest, Yahoo map, (2) rapid response, such as routing planning after an emergency event, and (3) operation planning, such as coordinating the super shuttle, or scheduling FedEx package pick up.

Future Directions

More than 80% of data collected are geospatial data. DGC is needed to integrate these datasets to support comprehensive applications from all walks of our life as envisioned by Vice President Gore.

The utilization of DGC to facilitate our daily life requires further research on (1) massive data management, such as Petabytes data archived by NASA, (2) intensive computing, such as real-time routing, (3) intelligent computing, such as real-time automatic identification of objects from in-situ sensors, (4) quality of services, such as an intelligent geospatial service searching engine, (5) interoperability, such as operational integration of DGC components in a real-time fashion, and (6) cyberinfrastructure, such as the utilization of massive desktops and other computing facilities to support intensive computing or massive data management.

Cross References

► Internet GIS
► Internet-Based Spatial Information Retrieval

Recommended Reading

1. Yang, C., Wong, D., Kafatos, M., Yang, R.: Network GIS. In: Qu, J., Gao, Y., Kafatos, M., Murphy, R., Solomonson, V. (eds.) Earth Science Satellite Remote Sensing, vol. 2, pp. 252–269. Springer and Tsinghua University Press, Beijing (2006)
2. ISO/IEC 13249. Information Technology-Database languages—SQL Multimedia and Application Packges, Part 3: Spatial. (2005)
3. ESRI. ArcXML Programmer's Reference Guide, ArcIMS 4. (2002)
4. Yang, C., Wong, D., Kafatos, M., Yang, R.: Implementing computing techniques to accelerate network GIS. In: Li, D., Xia, L. (eds.) Proc. SPIE Vol. 6418, 64181C, Geoinformatics 2006: GNSS and Integrated Geospatial Applications (2006)
5. Yang, C., Tao, V.: Distributed geospatial information service In: Rana, S., Sharma, J. (eds.) Frontiers of Geographic Information Technology, pp. 113–130. Springer, London (2005)
6. Plewe, B.: GIS Online: Information retrieval, Mapping, and the Internet. OnWord, Santa Fe (1997)
7. FGDC (1994) This website has information for clearinghouse, NSDI, GOS portal, and funding opportunities. http://www.fgdc.gov/. Accessed 20 Aug 2007
8. Open Geospatial Consortium (OGC) (1994) This website has information on WMS, WFS, WCS, CS-W, and other OGC specifications; http://www.opengeospatial.org
9. International Standards Organization (1994) Technical Committee 211: GeoInformation, Geomatics. http://www.isotc211.org/
10. Peng, Z.R., Tsou, M.H.: Internet GIS: Distributed Geographic Information Services for the Internet and Wireless Networks. Wiley, Hoboken, USA (2003)
11. Tao, V.: Microsoft Virtual Earth: New horizons in on-line mapping. Paper presented at GeoInformatics'2006, Wuhan, China, October 28–29 2006
12. Yang, C., Wong, D., Li, B.: Computing and computational issues of distributed GIS. Int. J. Geogr. Inf. Sci. **10**(1), 1–3 (2005)
13. Yang, C., Wong, D., Yang, R., Kafatos, M., Li, Q.: WebGIS performance improving techniques. Int. J. Geogr. Inf. Sci. **19**, 319–342 (2005)
14. Healey, R., Dowers, S., Gittings, B., Mineter, M.: Parallel Processing Algorithms for GIS. Taylor & Francis, Bristol, USA (1998)
15. Nolan, J.: An agent-based architecture for distributed imagery and geospatial computing PhD thesis, George Mason University (2003)
16. Joint Center for Intelligent Spatial Computing at George Mason University (2005) http://landscan.scs.gmu.edu:8080/. Accessed 20 Aug 2007
17. Pervasive Technology Labs at Indiana University (2004) http://grids.ucs.indiana.edu/ptliupages/. Accessed 20 Aug 2007
18. Executive Office of the President (EOP) (1994) Coordinating geographic data acquisition and access: the National Spatial Data Infrastructure. http://www.fgdc.gov/publications/documents/geninfo/execord.html. Accessed 20 Aug 2007
19. Mayer, L., Barbor, K., Boudreau, P., Chance, T., Fletcher, C., Greening, H., Li, R., Mason, C., Metcalf, K., Snow-Cotter, S., Wright, D.: A Geospatial Framework for the Coastal Zone: National Needs for Coastal Mapping and Charting. National Academies,Washington, DC (2004)
20. Cervone, G., Singh, R.P., Kafatos, M., Yu, C.: Wavelet maxima curves of surface latent heat flux anomalies associated with Indian earthquakes. Nat. Hazards Earth Syst. Sci. **5**, 87–99 (2005)
21. Birk R., Frederick, M., Dewayne, L.C., Lapenta, M.W.: NASA's Applied Sciences Program: transforming research results into operational success, Earth Imaging J. **3**(3), 18–23 (2006)
22. GEO (2005) The Global Earth Observation System of Systems (GEOSS) 10-Year Implementation Plan. http://earthobservations.org/docs/10-Year%20Implementation%20Plan.pdf.Accessed 20 Aug 2007

Distributed Geospatial Information Processing

► Distributed Geospatial Computing (DGC)

Distributed GIS

► Internet GIS

Distributed Hydrologic Modeling

BAXTER E. VIEUX
School of Civil Engineering and Environmental Science,
University of Oklahoma, Norman, OK, USA

Synonyms

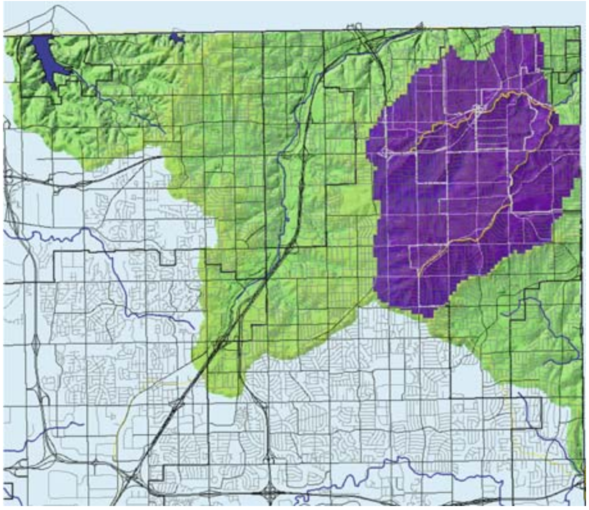
Spatial hydrologic modeling; GIS-based hydrology; Hydrogeology; Hydrology

Definition

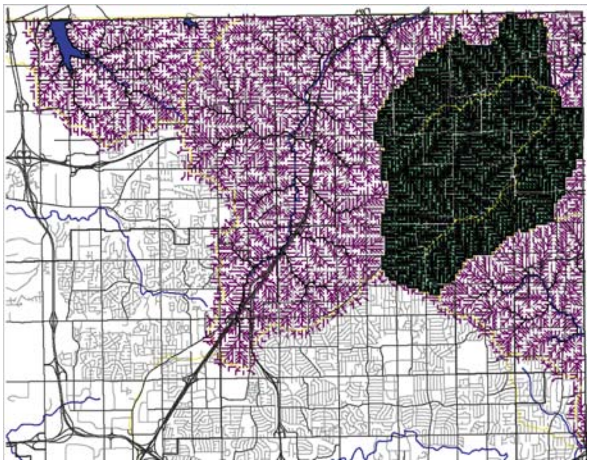
Distributed hydrologic modeling within a GIS framework is the use of parameter maps derived from geospatial data to simulate hydrologic processes. Distributed models of hydrologic processes rely on representing characteristics of the earth's surface that affect components of the water balance. Capturing the natural and human induced variability of the land surface at sufficient spatial resolution is a primary objective of distributed hydrologic modeling. Geospatial data is used to represent the spatial variation of watershed surfaces and subsurface properties that control hydrologic processes. Geospatial data is used in hydrologic modeling to characterize terrain, soils, land use/cover, precipitation, and meteorological parameters. The use of Geographic Information Systems is now commonplace in hydrologic studies. General purpose GIS software tools can be used for managing and processing spatial information for input to hydrologic models. Development of sophisticated GIS software and analysis tools, and the widespread availability of geospatial data representing digital terrain, land use/cover, and soils information have enabled the development of distributed hydrologic models.

Historical Background

Mathematical analogies used in hydrologic modeling rely on a set of equations and parameters that are representative of conditions within a watershed. Historical practice in hydrologic modeling has been to setup models using one value for each parameter per watershed area. When the natural variation of parameters representing infiltration, hydraulic roughness, and terrain slope are represented with a single parameter value, the resulting model is called a lumped model. Many such models, termed conceptual models, rely on regression or unit-hydrograph equations rather than the physics of the processes governing runoff, soil moisture, or infiltration. In a distributed modeling approach, a watershed is subdivided into grid cells or subwatersheds to capture the natural or human-induced variation of land surface characteristics. The smallest subdivision, whether a grid or subwatershed, is represented by a single value. Subgrid variability can be represented by a probabilistic distribution of parameter values. Figure 1 shows a subwatershed representation, whereas, Fig. 2 shows a gridded drainage network traced by finite elements laid out according to the principal flow direction. Whether a lumped or distributed approach to hydrologic modeling of a watershed is taken, GIS and geospatial data play an important role in characterizing the watershed characteristics. The application of GIS for lumped and distributed hydrologic modeling may be found in (Bedient et al., 2007).



Distributed Hydrologic Modeling, Figure 1 Subwatershed model representation



Distributed Hydrologic Modeling, Figure 2 Gridded representation with flow direction indicating the drainage network

Scientific Fundamentals

The gridded tessellation of geospatial data lends itself for use in solving equations governing surface runoff and other components of the hydrologic cycle. Hydrologic models may be integrated within a GIS, or loosely coupled outside of the GIS. Distributed modeling is capable of utilizing the geospatial data directly and with less averaging than lumped model approaches. A physics-based approach to distributed hydrologic modeling is where the numerical solution of conservation of mass, momentum, and energy is accomplished within the grid cells, which serve as computational elements in the numerical solution. Along the principal direction of land surface slope, the conservation of mass may be written for the overland flow depth, h, unit width flow rate, q, and the rainfall minus infiltration, $R - I$, as,

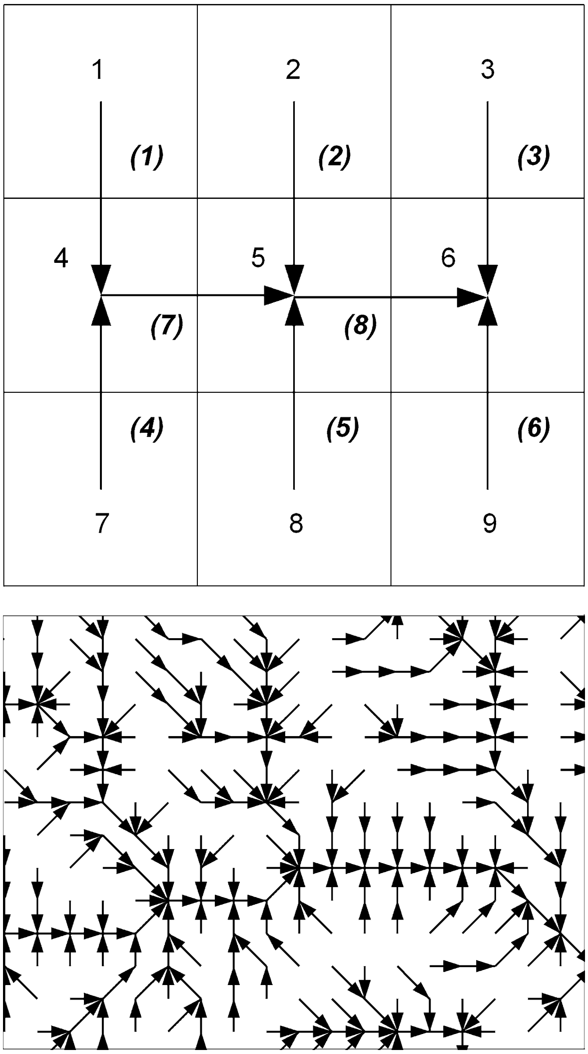
$$\frac{\partial h}{\partial t} + \frac{\partial q}{\partial x} = R - I \, . \tag{1}$$

The volumetric flow rate for a unit-width area, q, is the product of depth, h, and velocity, u, is related to the depth by a fully turbulent relationship such as the Manning Equation, which is,

$$u = \frac{c}{n} h^{5/3} s^{1/2} \tag{2}$$

where s, is the landsurface slope; n is the hydraulic roughness; and c is a constant depending on units. The slope is usually derived from the digital elevation model (DEM), and hydraulic roughness from land use/cover characteristics. Resampling from the resolution of the geospatial data to the model grid resolution is usually required. The grid used to solve the equations and the grid resolution of the geospatial data creates a linkage between the terrain characteristics and the equations governing the hydrologic process. The solution to these governing equations requires information from the DEM to define the direction of the principal gradient and the slope. The drainage network shown in Fig. 3 is composed of finite elements laid out in the direction of the principal land-surface slope.

Numerical solution of the governing equations in a physics-based model employs discrete elements. The three representative types of discrete solutions used are finite difference, finite element, and stream tubes (Moore, 1996). At the level of a computational element, a parameter is regarded as being representative of an average process. Averaging properties over a computational element used to represent the runoff process depends on the spatial variability of the physical characteristics. Maps of parameter values governing the mathematical solution are derived from geospatial data in a distributed model approach.



Distributed Hydrologic Modeling, Figure 3 Drainage network composed of finite elements in an elemental watershed (*top*), and a drainage network extracted for watershed defined by a DEM

Spatial Resolution

What spatial resolution should be selected for distributed hydrologic modeling? The required grid cell resolution or subdivision of a watershed depends on how well a single value can represent the hydrologic processes in a grid cell. Resolution should capture the broader scale variation of relevant features such as slope, soils, and land use/cover over the entire watershed. If a resolution that is too coarse is selected, the parameter and hydrologic processes can lose physical significance. Representing runoff depth in a grid cell that is 10-m, 100-m, or 1-km on a side can be used to model the results at the watershed outlet, but may not have physical significance locally at the grid cell subwatershed.

Changing the spatial resolution of data requires some scheme to aggregate parameter values at one resolution to another. Resampling is essentially a lumping process, which in the limit, results in a single value for the spatial domain. Resampling a parameter map involves taking the value at the center of the larger cell, averaging, or other operation. If the center of the larger cell happens to fall on a low/high value, then a large cell area will have a low/high value.

Over-sampling a parameter or hydrologic model input at finer resolution may not add any more information, either because the map, or the physical feature, does not contain additional information. Physical variations may be captured at a given resolution, however, there may be sub-grid variability that is not captured in the geospatial data. Dominant landuse classification schemes assign a single classification to a grid, yet may not resolve finer details or variations within the grid. The question of which resolution suffices for hydrologic purposes is answered in part by testing the quantity of information contained in a dataset as a function of grid resolution. Depending on the original data and resampling to coarser resolution, spatial detail may be lost. To be computationally feasible for hydrologic simulation, a model grid may need to be at coarser resolution than the digital terrain model. Resampling a digital elevation model to a coarser resolution dramatically decreases the slope derived from the coarser resolution DEM. Details on the effects of resolution on information content, and which resolution is adequate for capturing the spatial variability of the data may be found in (Vieux, 2004) and references contained therein.

Geospatial Data Deriving parameter values from remotely sensed or geospatial digital data requires reclassification and processing to derive useful input for a distributed hydrologic model. A brief description is provided below that relates geospatial data to distributed rainfall-runoff modeling.

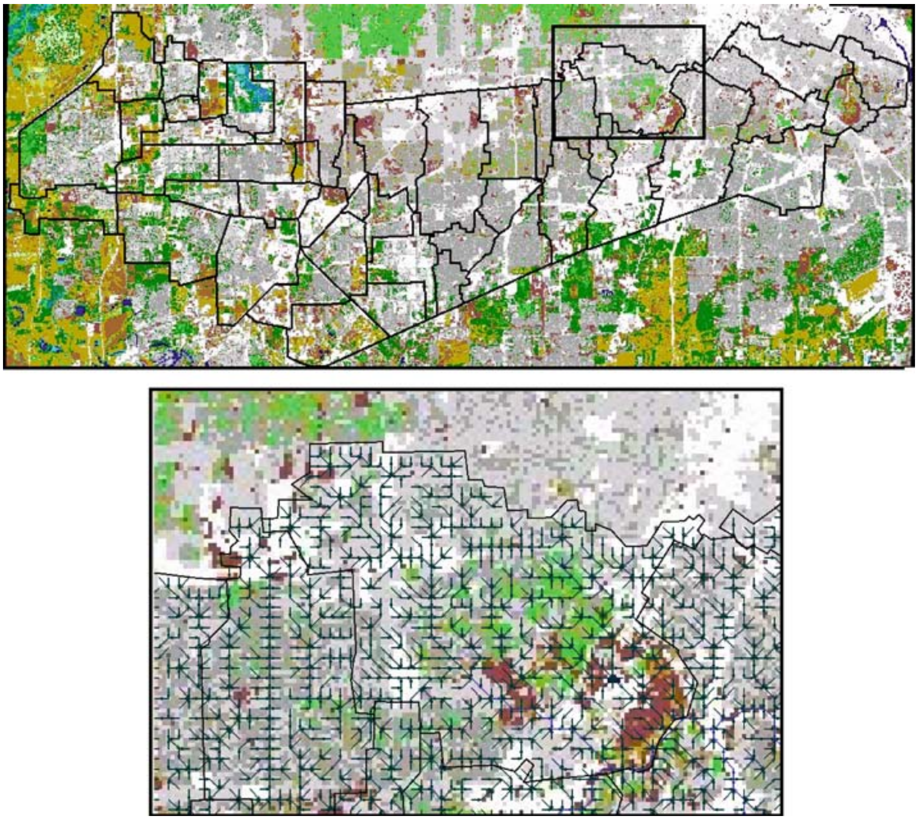
1. Soils/geologic material maps for estimating infiltration
   Soil maps provide information on physical properties of each soil mapping unit such as soil depth and layers, bulk density, porosity, texture classification, particle size distribution. These properties are used to derive hydraulic conductivity and other infiltration parameters. The polygon boundary of each mapping unit is reclassified then sampled into the model grid.
2. Digital Elevation Model (DEM)
   Delineation of watersheds and stream networks are accomplished using a DEM. Derivative maps of slope and drainage direction are the main input to the model for routing runoff through a gridded network model. Watershed delineation from the DEM using automated procedures is used to obtain the stream network and watershed boundary. Constraining the delineation with vector stream channels is useful in areas that are not well defined by the DEM. Depending on the resolution and the physical detail captured, a DEM may not contain specific hydrographic features such as channel banks, braided streams, or shallow depressions.
3. Land use/cover for estimating overland flow hydraulic parameters
   The land use and cover map is used to derive overland flow hydraulic properties and factors that modify infiltration rates. Pervious and impervious areas have dramatic influence on both runoff and infiltration rates. A lookup table must be provided based on published values or local experience to relate the landuse classification to hydraulic roughness. Hydraulic roughness can be developed by assigning a roughness coefficient to each landuse/cover classification in the map.
4. Channel cross-sections and hydraulic information
   Locational maps of channels are derived from the digital terrain model, stream networks digitized from high-resolution aerial photography, or derived from vector maps of hydrography. Other than channel location, the hydraulic and geometric characteristics must usually be supplied from sources other than GIS and commonly available geospatial data. Some channel hydraulic characteristics can be derived from aerial photography and geomorphic relationships that relate channel width to flow accumulation.

General purpose land use/cover classification schemes can be interpreted to provide initial parameter maps for use in modeling overland flow and model calibration (Vieux, 2004). Figure 4 shows such a map for Brays Bayou located in Houston Texas. The land use/cover classification derived from LandSat Thematic Mapper is reclassified into values of hydraulic roughness, n, and used in (2) to solve for runoff and to forecast flooding in the watershed (Vieux and Bedient, 2004).

Distributed Model Calibration

Parameter maps are used for setup and adjustment of the model to produce simulated response in agreement with observed quantities such as streamflow. Once the assembly of input and parameter maps for a distributed hydrologic model is completed, the model is usually calibrated or adjusted. Because parameter maps are derived from general purpose soils, land use/cover, and terrain, some adjustment is needed to calibrate the model to match observed flow. In the absence of observed flow, a physics-based model that has representative physical parameter values can produce useful results without calibration in ungauged watersheds.

Model calibration, implemented in a GIS, involves the adjustment of parameter maps to affect the value of the parameter, yet preserve the spatial patterns contained in the parameter map. Calibration may be performed man-



Distributed Hydrologic Modeling, Figure 4 Land use/cover classification derived from LandSat Thematic Mapper (*top*), and hydraulic roughness assigned to each finite element at the model grid resolution (*bottom*)

ually by applying scalar multipliers or additive constants to parameter maps until the desired match between simulated and observed is obtained. The ordered physics-based parameter adjustment (OPPA) method (Vieux and Moreda, 2003) is adapted to the unique characteristics of physics-based models. Predictable parameter interaction and identifiable optimum values are hallmarks of the OPPA approach that can be used to produce physically realistic distributed parameter values. Within a GIS context, physics-based models have the advantage that they are setup with parameters derived from geospatial data. This enables wide application of the model even where there is no gauge or only limited number of stream gauges available for model calibration. Parameter maps are adjusted to bring the simulated response into agreement with observations of streamflow. In the absence of observed streamflow, the model may be applied using physically realistic model parameters. The model calibration is accomplished by adjusting a map containing parameters or precipitation input. Let the set of values in a parameter map, $\{R\}$, be multiplied by a scalar, γ, such that,

$$\left\{R^*\right\} = \gamma \cdot \{R\} \tag{3}$$

where the resulting map, $\{R*\}$, contains the adjusted parameter values contained in the map. A similar procedure can be developed where the scalar in (2) is an additive constant that preserves the mean and the variance in the adjusted map (Vieux, 2004). If physically realistic values are chosen in the parameter map, then only small adjustments are needed to cause the model to agree with observed streamflow. Adjustments made to the other parameters in (1) and (2) affect the runoff volume and rate at any given location in the watershed. Hydraulic roughness derived from land use/cover, and hydraulic conductivity derived from soils maps are adjusted by a scalar adjustment that preserves the spatial variation.

Key Applications

Distributed hydrologic modeling within a GIS context is used for simulation of runoff, infiltration, soil moisture, groundwater recharge and evapotranspiration where the land surface is represented by geospatial data. This application supports hydrologic analysis and planning studies, and in operational forecasting of river flooding and stormwater.

Future Directions

A major advance in hydrology is accomplished using widely available geospatial data to setup a model. The availability of digital terrain, land use/cover, and soils makes it possible to setup the parameters for rainfall runoff modeling for any watershed. Remotely sensed data makes it feasible to create detailed watershed characteristics at high resolution. Terrain and land surface conditions can be derived from this high-resolution geospatial data to produce necessary watershed characteristics for hydrologic modeling. Future advances are anticipated in the use of distributed modeling in ungauged basins where streamflow is not available for calibration, but where geospatial data is available for model setup and application.

Cross References

► Hydrologic Impacts, Spatial Simulation

Recommended Reading

Bedient, P.B., Huber W.C., Vieux, B.E.: Hydrology and Floodplain Analysis, 4th edn. Prentice Hall, New Jersey, USA (2007)

DeBarry, P.A., Garbrecht, J., Garcia, L., Johnson, L.E., Jorgeson, J., Krysanova, V., Leavesley, G., Maidment, D., Nelson, E.J., Ogden, F.L., Olivera, F., Quimpo, R.G., Seybert, T.A., Sloan, W.T., Burrows, D., Engman, E.T.: GIS Modules and Distributed Models of the Watershed. American Society of Civil Engineers Water Resources Division – Surface Water Hydrology Committee, Reston, VA (1999), 120 pp.

Gurnell, A.M., Montgomery, D.R.: Hydrological Applications of GIS. John Wiley and Sons, New York (2000)

Jain, M.K., Kothyari, U.C., Ranga Raju, K.G.: A GIS Based Distributed Rainfall–runoff Model. J. Hydrol., **299**(1–2), pp. 107–135 (2004)

Maidment, D.R., and Djokic, D.: Hydrologic and Hydraulic Modeling Support with Geographic Information Systems. ESRI Press, Redlands, CA (2000)

Moore, I.D.: Hydrologic modeling and GIS. In: Goodchild, M.F., Steyaert, L.T., Parks, B.O., Johnston, C., Maidment, D.R., Crane, M.P., Glendinning, S. (eds.) GIS and Environmental Modeling: Progress and Research Issues, pp. 143–148. GIS World Books, Fort Collins, CO (1996)

Moore, I.D., Grayson, R.B., Ladson, A.R.: Digital Terrian Modeling: A Review of Hydrological, Geomorphological and Biological Applications. J. Hydrol. Process **5**(3–30) (1991)

Newell, C.J., Rifai, H.S., Bedient, P.B.: Characterization of Non-Point Sources and Loadings to Galveston Bay, 33 pp. Galveston Bay National Estuary Program (GBNEP-15), Clear Lake, Houston (1992)

Olivera, F., 2001. Extracting Hydrologic Information from Spatial Data for HMS Modeling." ASCE, J. Hydrol. Eng. 6(6): 524–530

Olivera, F., Maidment, D.: GIS-Based Spatially Distributed Model for Runoff Routing. Water Resources Research **35**(4):1155–1164 (1999)

Maidment, D.R., Morehouse, S. (eds.): Arc Hydro: GIS for Water Resources, 218 pp., ISBN 1-5894-8034-1. ESRI Press (2002)

Shamsi, U.: GIS Tools for Water, Wastewater, and Stormwater Systems, p. 392, ISBN: 0-7844-0573-5. American Society of Civil Engineers (ASCE Press), Reston Virginia (2002)

Singh, V.P., Fiorentino M.: Geographical Information Systems in Hydrology, Water Science and Technology Library, vol. 26, ISBN 0-7923-4226-7, Kluwer Academic Publishers, Norwell, Massachusetts (1996)

Safiolea, E., Bedient, P.B., Vieux, B.E.: Assessment of the Relative Hydrologic Effects of Land Use Change and Subsidence Using Distributed Modeling. In: Glenn E. Moglen (ed.) ASCE, Engineering, Ecological, and Economic Challenges Watershed. July 19–22, 2005, Williamsburg, Virginia, 178, 87 (2005)

Tate, E., Maidment, D., Olivera, F., Anderson, D.: Creating a Terrain Model for Floodplain Mapping, ASCE. J. Hydrol. Eng., **7**(2):100–108 (2002)

Vieux, B.E.: DEM Aggregation and Smoothing Effects on Surface Runoff Modeling. J. Comput. Civ. Eng. **7**(3), pp. 310–338 (1993)

Vieux, B.E.: Distributed Hydrologic Modeling Using GIS. Water Science Technology Series, vol. 48, 2nd edn. Kluwer Academic Publishers, Norwell, Massachusetts. ISBN 1-4020-2459-2, p. 289. CD-ROM including model software and documentation (2004)

Wilson, J.P. (ed.), Gallant, J.C.: Terrain Analysis: Principles and Applications, p. 512, ISBN 0-471-32188-5. John Wiley & Sons, New Jersey, USA (2000)

Vieux, B.E., Moreda F.G.: Ordered Physics-Based Parameter Adjustment of a Distributed Model. In: Duan Q., Sorooshian S., Gupta H.V., Rousseau A.N., Turcotte, R. (eds.) Advances in Calibration of Watershed Models. Water Science and Application Series, vol. 6, pp. 267–281. American Geophysical Union, Washington, USA (2003)

Vieux, B.E., Bedient P.B.: Assessing urban hydrologic prediction accuracy through event reconstruction. Special Issue on Urban Hydrology. J. Hydrol. **299**(3–4), 217–236 (2004)

Distributed Information Systems

► Pandemics, Detection and Management

Distributed Localization

► Localization, Cooperative

Distribution Logistics

► Routing Vehicles, Algorithms

Divide and Conquer

► Skyline Queries

DLG

► Spatial Data Transfer Standard (SDTS)

Document Object Model

► Scalable Vector Graphics (SVG)

Driving Direction

► Fastest-Path Computation

Dual Space-Time Representation

► Indexing Schemes for Multi-dimensional Moving Objects

Dynamic Generalization

► Generalization, On-the-Fly

Dynamic Travel Time Maps

SOTIRIS BRAKATSOULAS[1], DIETER PFOSER[1], NECTARIA TRYFONA[2], AGNÈS VOISARD[3]
[1] RA Computer Technology Institute, Athens, Greece
[2] Talent SA, Athens, Greece
[3] Fraunhofer ISST and FU Berlin, Berlin, Germany

Synonyms

Spatio-temporal data warehousing; Travel time computation; Spatial Causality; Characteristic Travel Time

Definition

The application domain of intelligent transportation is plagued by a shortage of data sources that adequately assess traffic situations. Currently, to provide routing and navigation solutions, an unreliable travel time database that consists of static weights as derived from road categories and speed limits is used for road networks. With the advent of floating car data (FCD) and specifically the GPS-based tracking data component, a means was found to derive accurate and up-to-date travel times, i. e., qualitative traffic information. FCD is a by-product in fleet management applications and given a minimum number and uniform distribution of vehicles, this data can be used for accurate traffic assessment and also prediction. Map-matching the tracking data produces travel time data related to a specific edge in the road network. The *dynamic travel time map* (DTTM) is introduced as a means to efficiently supply dynamic weights that are derived from a collection of historic travel times. The DTTM is realized as a spatiotemporal data warehouse.

Historical Background

Dynamic travel time maps were introduced as an efficient means to store large collections of travel time data as produced by GPS tracking of vehicles [6]. Dynamic travel time maps can be realized using standard data warehousing technology available in most commercial database products.

Scientific Fundamentals

A major accuracy problem in routing solutions exists due to the unreliable travel time associated with the underlying road network.

A road network is modeled as a directed graph $G = (V, E)$, whose vertices V represent intersections between links and edges E represent links. Additionally, a real-valued weight function $w : E \to \mathbf{R}$ is given, mapping edges to weights. In the routing context, such weights typically correspond to speed types derived from road categories or based on average speed measurements. However, what is important is that such weights are static, i. e., once defined they are rarely changed. Besides, such changes are rather costly as the size of the underlying database is in the order of dozens of gigabytes.

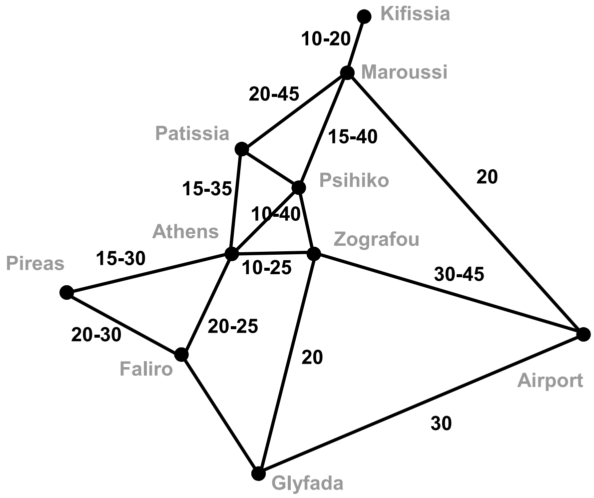
Although the various algorithmic solutions for routing and navigation problems [3,7] are still subject to further research, the basic place for improving solutions to routing problems is this underlying weight-based database $DB(w(u,v))$.

The DTTM will be a means to make the weight database fully dynamic with respect to not only a spatial (what road portion) but also a temporal argument (what time of day). The idea is to derive dynamic weights from collected FCD. Travel times and thus dynamic weights are derived from FCD by relating its tracking data component to the road network using map-matching algorithms [1]. Using the causality between historic and current traffic conditions, weights – defined by temporal parameters – will replace static weights and will induce impedance in traversing some links. Figure 1 gives an example of fluctuating travel times in a road network.

Travel Time Causality in the Road Network

The collection of historical travel time data provides a strong basis for the derivation of dynamic weights provided that one can establish the causality of travel time with respect to time (time of the day) and space (portion of the road network) [2,9].

Temporal causality establishes that for a given path, although the travel time varies with time, it exhibits a reoccurring behavior. An example of temporal causality is shown in Fig. 2a. For the same path, two travel time pro-



Dynamic Travel Time Maps, Figure 1 Fluctuating travel times (in minutes) in a road network example (Athens, Greece)

files, i. e., the travel time varying with the time of the day, are recorded for different weekdays. Although similar during nighttime and most daytime hours, the travel times differ significantly during the period of 16 to 22 h. Here, on one of the days the shops surrounding this specific path were open from 17 to 21 h in the afternoon.

Spatial causality establishes that travel times of different edges are similar over time. An example is given in Fig. 2b. Given two different paths, their travel time profiles are similar. Being in the same shopping area, their travel time profile is governed by the same traffic patterns, e. g., increased traffic frequency and thus increased travel times from 6 to 20 h, with peaks at 8 h and around noon.

Overall, discovering such temporal and spatial causality affords hypothesis testing and data mining on historic data sets. The outcome is a set of rules that relate (cluster) travel times based on parts of the road network and the time in question. A valid hypothesis is needed that selects historic travel time values to compute meaningful weights.

Characteristic Travel Times = Aggregating Travel Times

A problem observed in the example of Fig. 2b is that travel times, even if causality is established, are not readily comparable. Instead of considering absolute travel times that relate to specific distances, the notion of relative travel time ρ is introduced, which for edge e is defined as follows:

$$\rho(e) = \frac{\tau(e)}{l(e)}\,, \qquad (1)$$

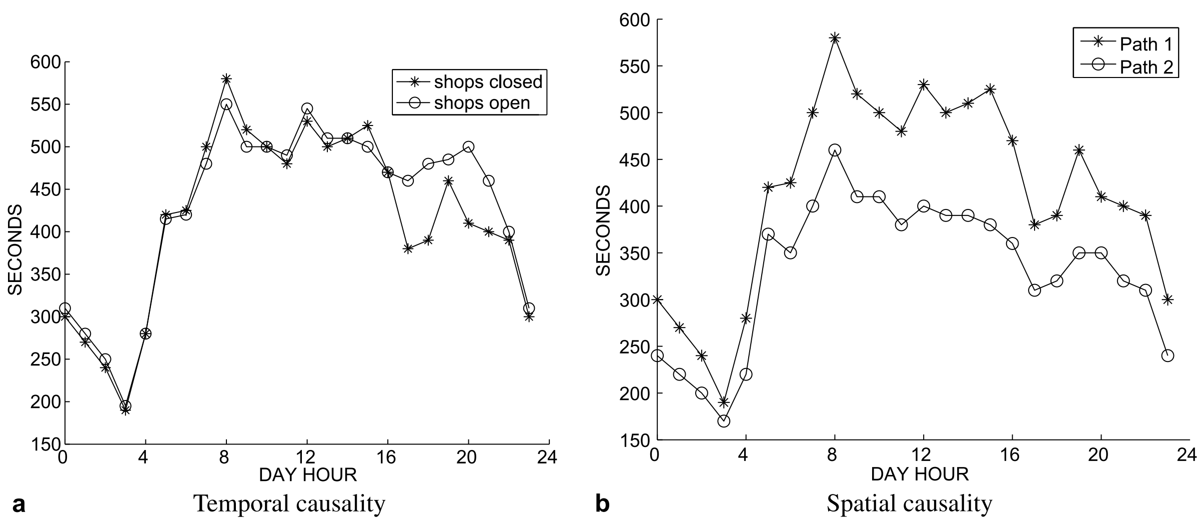
where $\tau(e)$ and $l(e)$ are the recorded travel time and edge length, respectively.

Given a set of relative travel times $P(e)$ related to a specific network edge e and assuming that these times are piecewise independent, the *characteristic travel time* $\chi(P)$ is defined by the triplet cardinality, statistical mean, and variation as follows,

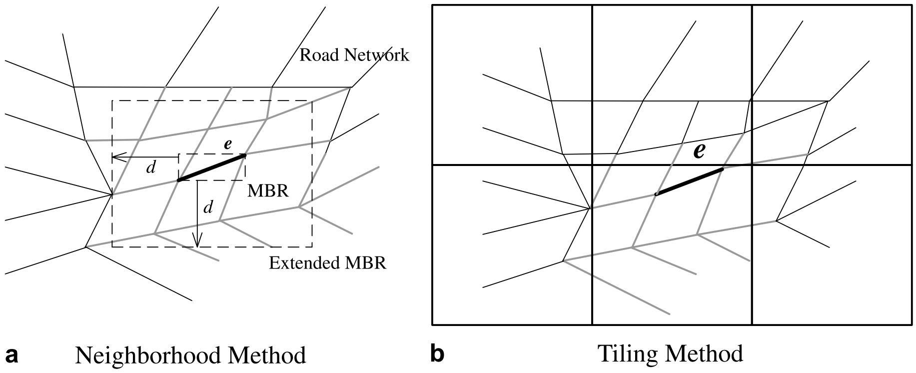
$$\chi(P) = \{|P|, E[P], V[P]\}\,, \qquad (2)$$

$$E[P] = \sum_{\rho \in P} \frac{\rho}{|P|}\,, \qquad (3)$$

$$V[P] = \sum_{\rho \in P} \frac{(\rho - E[P])^2}{|P|}\,. \qquad (4)$$



Dynamic Travel Time Maps, Figure 2 Relating travel times



Dynamic Travel Time Maps, Figure 3 Characteristic travel time computation methods

The critical aspect for the computation of $P(e)$ is the set of relative travel times selected for the edge e in question based on temporal and spatial inclusion criteria $I_T(e)$ and $I_S(e)$, respectively.

$$P(e) = \{\rho(e^*, t) : e^* \in I_S(e) \wedge t \in I_T(e)\} . \quad (5)$$

$I_S(e)$, a set of edges, typically contains the edge e itself, but can be enlarged, as seen later on, to include further edges as established by an existing spatial causality between the respective edges. $I_T(e)$, a set of time periods, is derived by existing temporal causality. The characteristic travel time essentially represents a dynamic weight, since depending on a temporal inclusion criterion (e. g., time of the day, day of the week, month, etc.), its value varies.

FCD and thus travel times are not uniformly distributed over the entire road network, e. g., taxis prefer certain routes through the city. To provide a dynamic weight database for the entire road network, a considerable amount of FCD is needed on a per edge basis, i. e., the more available data, the more reliable will be the characteristic travel time.

While it is possible to compute the characteristic travel times for frequently traversed edges only based on data related to the edge in question, for the non-frequently traversed edges, with their typical lack of data, complementary methods are needed. The simplest approach is to substitute travel times by static link-based speed types as supplied by map vendors. However, following the spatial causality principle, the following three prototypical methods can be defined. The various approaches differ in terms of the chosen spatial inclusion criterion $I_S(e)$, with each method supplying its own approach.

Simple Method. Only the travel times collected for a specific edge are considered. $I_S(e) = \{e\}$.

Neighborhood Method. Exploiting spatial causality, i. e., the same traffic situations affecting an entire area, a simple neighborhood approach is used by considering travel times of edges that are (i) contained in an enlarged minimum bounding rectangle (MBR) around the edge in question and (ii) belong to the same road category. Figure 3a shows a network edge (bold line) and an enclosing MBR (small dashed rectangle) that is enlarged by a distance d to cover the set of edges considered for travel time computation (thicker gray lines). $I_S(e) = \{e^* : e^*$ contained_in d-expanded MBR$(e) \wedge L(e^*) = L(e)\}$, with $L(e)$ being a function that returns the road category of edge e.

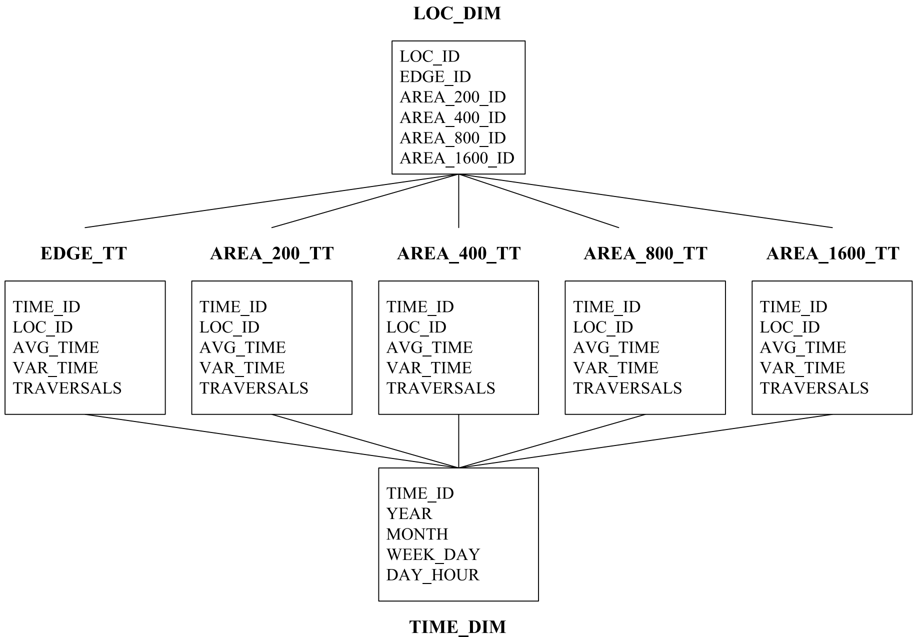
Tiling Method. Generalizing the neighborhood method, a fixed tiling approach for the network is used to categorize edges into neighborhoods. It effectively subdivides the space occupied by the road network into equally sized tiles. All travel times of edges belonging to the same tile and road category as the edge in question are used for the computation of the characteristic travel time. Figure 3b shows a road network and a grid. For the edge in question (bold line) all edges belonging to the same tile (thicker gray lines) are used for the characteristic travel time computation. $I_S(e) = \{e^* : e^* \in Tile(e) \wedge L(e^*) = L(e)\}$

Both the neighborhood and the tiling method are effective means to compensate for missing data when computing characteristic travel times. Increasing the d in the neighborhood method, increases the number of edges and thus the potential number of relative travel times considered. To achieve this with the tiling method, the tile sizes have to be increased.

Dynamic Travel Time Map

The basic requirement to the DTTM is the efficient retrieval of characteristic travel times and thus dynamic weights on a per-edge basis. Based on the choice of the various travel time computation methods, the DTTM needs to support each method in kind.

The travel time computation methods use arbitrary temporal and spatial inclusion criteria. This suggests the use of a data warehouse with relative travel times as a data warehouse fact and space and time as the respective dimensions. Further, since the tiling method proposes regular subdivisions of space, one has to account for a potential



Dynamic Travel Time Maps, Figure 4 Data warehouse schema

lack of travel time data in a tile by considering several subdivisions of varying sizes.

The multidimensional data model of the data warehouse implementing the DTTM is based on a star schema. Figure 4 shows the schema comprising five fact tables and two data warehouse dimensions. The two data warehouse dimensions relate to time, TIME_DIM, and to space, LOC_DIM, implementing the respective granularities as described in the following.

Spatial subdivisions of varying size can be seen as subdivisions of varying granularity that form a dimensional hierarchy. These subdivisions relate to the tiling method used for characteristic travel time computation. A simple *spatial hierarchy* with quadratic tiles of side length 200, 400, 800, and 1600 meters respectively is adopted, i. e., four tiles of x m side-length are contained in the corresponding greater tile of $2x$ m side-length. Consequently, the spatial dimension is structured according to the hierarchy *edge*, *area_200*, *area_400*, *area_800*, *area_1600*. Should little travel time data be available at one level in the hierarchy, one can consider a higher level, e. g., *area_400* instead of *area_200*.

Obtaining characteristics travel times means to relate individual travel times. Using an underlying temporal granularity of *one hour*, all relative travel times that were recorded for a specific edge during the same hour are assigned the same timestamp. The temporal dimension is structured according to a simple hierarchy formed by the *hour of the day*, 1 to 24, with, e. g., 1 representing the time from 0:00 am to 1:00 am, the *day of the week*, 1 (Monday) to 7 (Sunday), *week*, the calendar week, 1 to 52, *month*, 1 (January) to 12 (December), and *year*, 2000–2003, the years for which tracking data was available to us.

The measure that is stored in the *fact tables* is the characteristic travel time χ in terms of the triplet $\{|P|, E[P], V[P]\}$. The fact tables comprise a base fact table EDGE_TT and four derived fact tables, AREA_200_TT, AREA_400_TT, AREA_800_TT, and AREA_1600_TT, which are aggregations of EDGE_TT implementing the spatial dimension hierarchy. Essentially, the derived fact tables contain the characteristic travel time as computed by the tiling method for the various extents.

In *aggregating travel times* along the spatial and temporal dimensions, the characteristic travel time $\chi(C) = \{|C_i|, E[C_i], V[C_i]\}$ for a given level of summarization can be computed based on the respective characteristic travel times of a lower level, $\chi(S_j)$, without using the initial set of characteristic travel times P as follows.

$$|C_i| = \sum_{S_j \in C_i} |S_j| \tag{6}$$

$$E[C_i] = \frac{\sum_{S_j \in C_i} |S_j| \cdot E[S_j]}{|C_i|} \tag{7}$$

$$V[C_i] = \frac{\sum_{S_j \in C_i} |S_j| \left(V(S_j) + E[S_j]\right)}{|C_i|} - E^2[C_i]\,. \tag{8}$$

Implementation and Empirical Evaluation

To evaluate its performance, the DTTM was implemented using an Oracle 9i installation and following the data warehouse design “best practices” described in [5,8]. The primary goal of these best practices is to successfully achieve the adoption of the “star transformation” query processing scheme by the Oracle optimizer. The latter is a special query processing technique, which gives fast response times when querying a star schema.

The steps involved in achieving an optimal execution plan for a star schema query require the setup of an efficient indexing scheme and some data warehouse-specific tuning of the database (proper setting of database initialization parameters and statistics gathering for use by the cost-based optimizer of Oracle). The indexing scheme is based on bitmap indexes, an index type that is more suitable for data warehousing applications compared to the traditional index structures used for OLTP systems.

Using this DTTM implementation, experiments were conducted to confirm the accuracy of the travel time computation methods and to assess the potential computation speed of dynamic weights. The data used in the experiments comprised roughly 26000 trajectories that in turn consist of 11 million segments. The data was collected using GPS vehicle tracking during the years 2000 to 2003 in the road network of Athens, Greece. Details on the experimental evaluation can be found in [6].

To assess the relative accuracy of the travel time computation methods, i. e., simple method vs. neighborhood and tiling method, three example paths of varying length and composition (frequently vs. non-frequently traversed edges) were used. The simple method was found to produce the most accurate results measured in terms of the standard deviation of the individual travel times with respect to computed characteristic travel time. The tiling method for small tile sizes (200×200 m) produces the second best result in terms of accuracy at a considerably lower computation cost (number of database I/O operations). Overall, to improve the travel time computation methods in terms of accuracy, a more comprehensive empirical study is needed to develop appropriate hypothesis for temporal and spatial causality between historic travel times.

To evaluate the feasibility of computing dynamic weights, an experiment was designed to compare the computation speed of dynamic to static weights. To provide static weights, the experiment utilizes a simple schema consisting of only one relation containing random-generated static weights covering the entire road network. Indexing the respective edge ids allows for efficient retrieval. In contrast, dynamic weights are retrieved by means of the DTTM using the tiling method. The query results in each case comprise the characteristic travel time for the edge in question. The results showed that static weights can be computed *nine* times faster than dynamic weights. Still, in absolute terms, roughly 50 dynamic weights can be computed per second. Using further optimization, e. g., in routing algorithms edges are processed in succession, queries to the DTTM can be optimized and the number of dynamic weights computed per time unit can be increased further.

Summary

The availability of an accurate travel time database is of crucial importance to intelligent transportation systems. *Dynamic travel time maps* are the appropriate data management means to derive dynamic – in terms of time and space – weights based on collections of large amounts of vehicle tracking data. The DTTM is essentially a spatio-temporal data warehouse that allows for an arbitrary aggregation of travel times based on spatial and temporal criteria to efficiently compute characteristic travel times and thus dynamic weights for a road network. To best utilize the DTTM, appropriate hypotheses with respect to the spatial and temporal causality of travel times have to be developed, resulting in accurate characteristic travel times for the road network. The neighborhood and the tiling method as candidate travel time computation methods can be seen as the basic means to implement such hypotheses.

Key Applications

The DTTM is a key data management construct for algorithms in intelligent transportation systems that rely on a travel time database accurately assessing traffic conditions. Specific applications include the following.

Routing

The DTTM will provide a more accurate weight database for routing solutions that takes the travel time fluctuations during the day (daily course of speed) into account.

Dynamic Vehicle Routing

Another domain in which dynamic weights can prove their usefulness is dynamic vehicle routing in the context of managing the distribution of vehicle fleets and goods. Traditionally, the only dynamic aspects were customer orders. Recent literature, however, mentions the traffic conditions, and thus travel times, as such an aspect [4].

Traffic Visualization, Traffic News

Traffic news relies on up-to-date traffic information. Combining the travel time information from the DTTM with current data will provide an accurate picture for an entire geographic area and road network. A categorization of the respective speed in the road network can be used to visualize traffic conditions (color-coding).

Future Directions

An essential aspect for providing accurate dynamic weights is the appropriate selection of historic travel times. To provide and evaluate such hypothesis, extensive data analysis is needed possibly in connection with field studies.

The DTTM provides dynamic weights based on historic travel times. An important aspect to improve the overall quality of the dynamic weights is the integration of current travel time data with DTTM predictions.

The DTTM needs to undergo testing in a scenario that includes massive data collection and dynamic weight requests (live data collection and routing scenario).

Cross References

► Floating Car Data
► Map-Matching

Recommended Reading

1. Brakatsoulas, S., Pfoser, D., Sallas, R., Wenk, C.: On map-matching vehicle tracking data. In: Proc. 31st VLDB conference, pp. 853–864 (2005)
2. Chen, M., Chien, S.: Dynamic freeway travel time prediction using probe vehicle data: link-based vs. path-based. J. Trans. Res. Board **1768**, 157–161 (2002)
3. Dechter, R., Pearl, J.: Generalized best-first search strategies and the optimality of A*. J. ACM **32**(3), 505–536 (1985)
4. Fleischmann, B., Sandvoß, E., Gnutzmann, S.: Dynamic vehicle routing based on on-line traffic information. Trans. Sci. **38**(4), 420–433 (2004)
5. Kimball, R.: The Data Warehouse Toolkit, 2nd edn. Wiley, New York (2002)
6. Pfoser, D., Tryfona, N., Voisard, A.: Dynamic travel time maps – enabling efficient navigation. In: Proc. 18th SSDBM Conference, pp. 369–378 (2006)
7. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Prentice Hall, New York (2003)
8. Scalzo, B.: Oracle DBA Guide to Data Warehousing and Star Schemas, 1st edn. Prentice Hall, New York (2003)
9. Schaefer, R.-P., Thiessenhusen, K.-U., Wagner, P.: A traffic information system by means of real-time floating-car data. In: Proc. ITS World Congress, Chicago (2002)

Dynamics

► Geographic Dynamics, Visualization And Modeling

Early Warning

► Hotspot Detection, Prioritization, and Security

Earth Observation

► Evolution of Earth Observation

Earth Observation Standards

► Standards, Critical Evaluation of Remote Sensing

Ecological Planning and Modeling

► Environmental Planning and Simulation Tools

Edge Routing Problems

► Graph Theory, Konigsberg Problem

Egenhofer Operators

► Dimensionally Extended Nine-Intersection Model (DE-9IM)

Egg-Yolk Calculus

► Representing Regions with Indeterminate Boundaries

Egg-Yolk Model

► Objects with Broad Boundaries

Electromagnetic Tagging

► Radio Frequency Identification (RFID)

Electronic Atlases

► Web Mapping and Web Cartography

Electronic Identification

► Radio Frequency Identification (RFID)

Elevation Reference Surface (Datum)

► Photogrammetric Products

Embodiment, Individualization

► Geospatial Semantic Web: Personalisation

Emergency Evacuation, Dynamic Transportation Models

YI-CHANG CHIU[1], HENRY X. LIU[2]
[1] Department of Civil Engineering and Engineering Mechanics, University of Arizona, Tucson, AZ, USA
[2] Department of Civil Engineering, University of Minnesota, Minneapolis, MN, USA

Synonyms

Decision support tools for emergency evacuations; Emergency evacuation models

Definition

Emergency evacuation models are computerized tools that explicitly represent the time-varying disaster trajectories, evacuation (persons or goods) traffic flows as well as change of location of evacuees. These dynamic models generally involve vehicular traffic representation at varying degrees of resolutions, ranging from a coarse macroscopic analytical flow approximation to microscopic vehicle-based traffic simulation.

The integral role that geographic information systems (GIS) plays in emergency evacuation management is in that GIS have the ability to overlay mapping data with location identifiers for specific objects and factors. In emergency evacuation applications, GIS can incorporate mapping information and organize critical model building information such as evacuee spatial distributions and socio-demographic as well as the geometric configurations of evacuation routes.

GIS are also powerful in organizing and processing evacuation model outputs based on GIS's spatial analysis capabilities. GIS enable a quick translation of data among different models with diversified format requirements. GIS also permit a visually effective display of model outputs, allowing quicker and effective decision-making in both offline (planning) and real-time (operations) contexts.

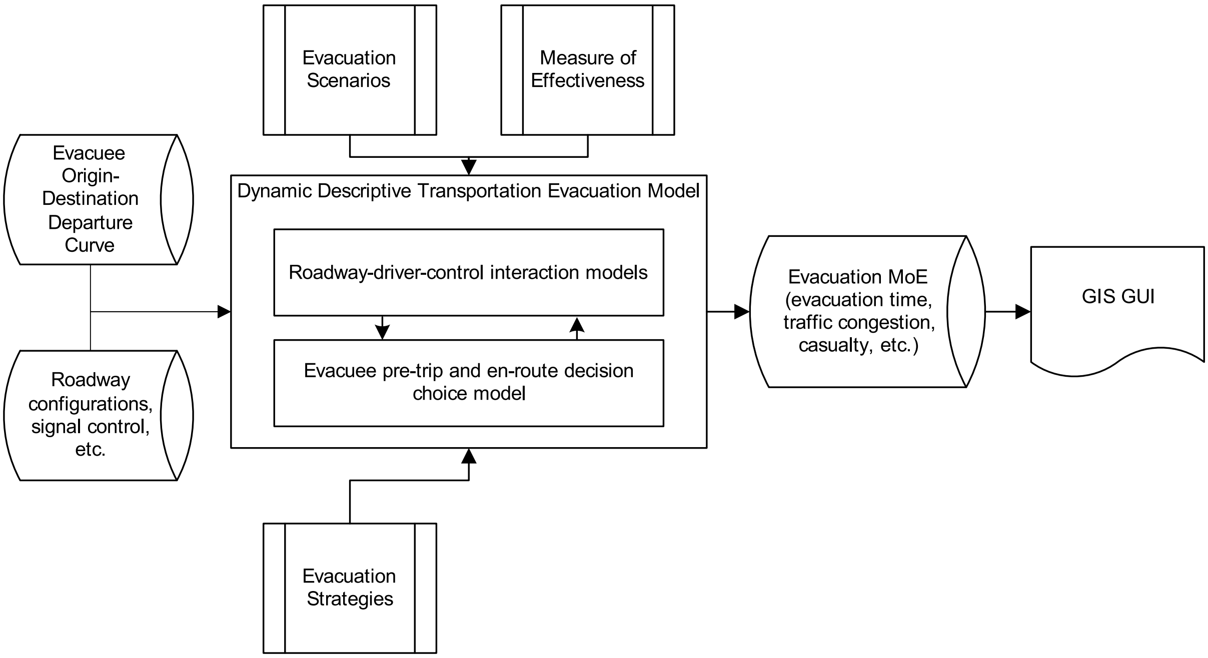
Historical Background

The majority of emergency response research started in the wake of the Three Mile Island nuclear incident in Pennsylvania, U.S. in 1979. Even though no immediate deaths or injuries of plant workers or members of the nearby community were reported, this incident brought about sweeping regulatory changes in emergency response practices as well as elevated research activities in this area [1,2,3,4,5,6]. Since then, researchers have also studied emergency response related to other natural disasters such as hurricanes, earthquakes, or wildfires [7,8,9,10,11,12,13]. Hurricanes, excluding Hurricanes Katrina and Rita in 2005, have caused nearly 100 billion dollars of damage to the U.S. since the early 1990's. Preliminary estimation shows that Katrina alone has caused over 150 billion dollars of damage to the U.S. economy [14]. Terrorist activities have a long history, but the level of devastation that they have created in recent years has called for a large-scale revision of homeland security regulations and emergency response practices in the U.S. and around the world [15,16,17,18,19]. Several recent studies have started to address the unique issues related to no-notice extreme events [20,21,22,23]. Nowadays, emergency response agencies in most of the U.S. cities rely on emergency response plans or computerized models in times of major disasters. However, no-notice disasters requiring mass evacuation continue to be one of the most devastatingly challenging issues faced by emergency response agencies.

Most existing plans or models allow the evacuation planners to evaluate and plan for "what-if" scenarios. When a disaster occurs, the best-fit plan may be selected to assist with evacuation operations, so evacuation plans are being used mainly as a reference strategy in the event of actual evacuation. Actual evacuation operations rely primarily on emergency officers' experience and judgment. Models capable of assisting the emergency planner or operator in better understanding the outcomes of various disaster scenarios and evacuation strategies have been attempted by researchers using various modeling techniques ranging from GIS-based static spatial analysis [24,25,26,27] to simulation-based microscopic traffic analysis.

Over the decades, in the pursuit of dynamic modeling for evacuation modeling, several legacy models are being improved from a static planning model to a dynamic real-time decision tool. Examples of these models include the followings: I-DYNEV is a macroscopic simulation model that was developed in the early 1980's to model evacuation from nuclear sites [3]. MASSVAC uses macroscopic traffic flow models to forecast hurricane evacuation performance of Virginia Beach [7,28]. The Oak Ridge Evacuation Modeling System (OREMS) is aimed at dynamic modeling of evacuation from a variety of disasters [29] based on macroscopic traffic flow approximation. Some recent dynamic traffic assignment (DTA)-based traffic estimation and prediction models such as DYNASMART-P [30] and DynaMIT [31] have the potential to be utilized for the real-time emergency response due to their dynamic network modeling capability. The mesoscopic traffic simulation model based approach also explicitly incorporates routing and tripmakers' responses to information. Instead of simulation, some recent studies apply the cell transmission model as the traffic modeling mechanism for evacuation [32,33,34]. Overall, these dynamic models usually require evacuation origin-destination data as model inputs and these data (particularly the destination information) are usually unknown at the time of emergency. Some of the above models have the representation of multiple user class with different routing objectives, but no treatment is given to user classes with different priorities.

Research relating to post-disaster relief is relatively sparse in literature. Sherali et al. [35] considers the shelter location-allocation decisions for hurricane evacuation. The model selects a set of candidate shelters from among a given set of admissible alternatives in a manner feasible to available resources and prescribes an evacuation plan which minimizes the total congestion-related evacuation time. The traffic flow modeling however is a rather simplified approximation which may raise the concern of not being able to realistically represent traffic in an actual city traffic network [35]. A comprehensive emergency logistic study was done by Hagani and Oh [36], which proposed a formulation and solution algorithms for



Emergency Evacuation, Dynamic Transportation Models, Figure 1 General modeling framework for descriptive evacuation models

a multi-commodity, multi-modal logistical problem in disaster relief management. Chiu and Zheng [34] also proposed a modeling approach to assign routes for evacuee groups with different priorities. Brown and Valliliou [37] developed a real-time decision support system which uses optimization methods, simulation and the decision maker's judgment for operational assignment of units to tasks and for tactical allocation of units to task requirements in repairing major damage to public works following a disaster.

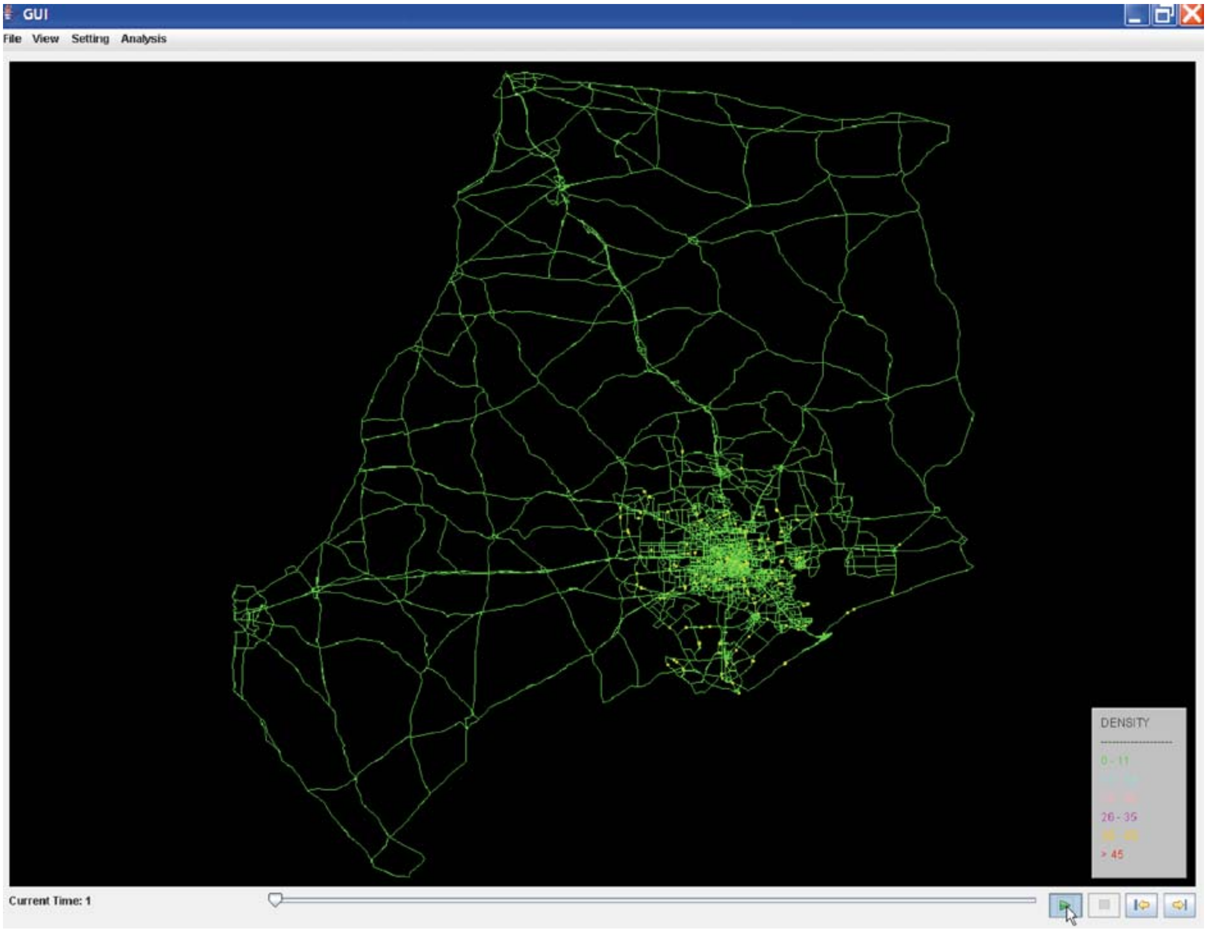
Scientific Fundamentals

In general, emergency evacuation models for transportation planning and operations can be classified into two distinct categories; namely, descriptive and prescriptive models. While descriptive models represent how individual evacuees travel through a given transportation system and the collective traffic patterns resulting from individual choices, prescriptive models are aimed at seeking control policies to guide the evacuation traffic toward certain objectives desired by emergency managers. Therefore, descriptive models are mainly used for the purpose of transportation planning under emergency evacuations in order to evaluate the performance of alternative evacuation strategies. Prescriptive models, on the other hand, are particularly suitable for the purpose of traffic operations, to explicitly search for optimal evacuation strategies. More details are offered in the following sections.

Descriptive Emergency Evacuation Models

The descriptive transportation evacuation planning models are aimed at evaluating pre-defined evacuation strategies without the automatic and explicit search of optimal evacuation strategies. As depicted in Fig. 1, the inputs of such models generally include the estimates of evacuees' evacuation origins, destinations, and departure times. The transportation network encompassing the evacuation boundary, including hot zones and safe destinations, needs to be prepared. Explicit modeling of intersection signals is optional, with a certain equivalent intersection capacity or delay estimates as alternative inputs. Another set of input for the models includes user-defined evacuation scenarios, measure of effectiveness (MoE) and evacuation strategies. The evacuation scenarios may encompass a wide range of disaster scenarios that differ in terms of available lead times (e. g., no-noticed vs. short-notice scenarios), stationary or moving disaster trajectories (earthquake vs. hurricanes, flooding or airborne hazmat dispersion), or natural or man-made disasters. The evacuation operational or planning strategies are generally devised by the evacuation operators or planners, which may entail the configuration of contra-flow lanes, phased evacuations, location and distribution of shelter or medical facilities or low-mobility communities, etc.

The dynamic descriptive emergency evacuation models take all such inputs and model (simulate) the vehicular traffic flow dynamics through the internal simulation/

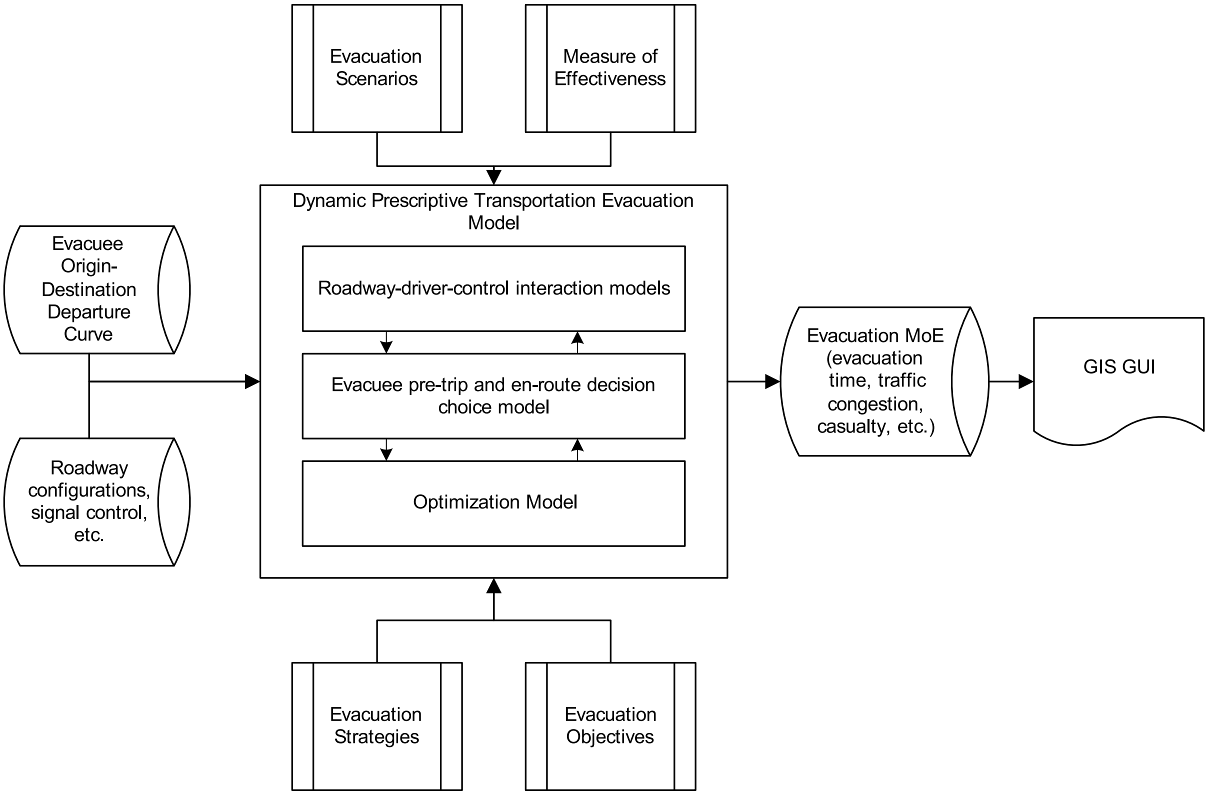


Emergency Evacuation, Dynamic Transportation Models, Figure 2 Simulation of central texas evacuation in MALTA [43]

modeling logics that capture the roadway-driver-control interactions. Depending on the granularity and design of the modeling details, some dynamic models may be capable of explicitly modeling the routing behaviors of individual evacuees and their evacuation travel behaviors in response to information, control measures and congestion. The user-defined measures of effectiveness (MoE) are generated by the models and the user can usually visualize the results from the user graphical interface (mostly GIS-based GUI) as well as the text-based report. Selected known dynamic models that have been applied in evacuation modeling include I-DYNEV [38], DYNASMART-P [39,40], DYNAMIT [41], TSIS [42] and MALTA [43]. Figure 2 shows the simulation snapshot using MALTA to simulate the evacuation traffic in Central Texas.

Prescriptive Emergency Evacuation Models

The major distinction between the prescriptive and the descriptive models are in that the prescriptive models enable the automatic and explicit elucidation and search of optimal evacuation strategies based on the user-defined objective(s) [34,44]. To cope with the dynamic and uncertain nature of evacuation traffic, continuous system monitoring capability is required and system optimization may need to be performed iteratively in order to adapt to the changing environment [45]. One particular close-loop control, model reference adaptive control (MRAC), emphasizes monitoring system performance by taking measurements from the real world traffic [46]. The MRAC evaluates the current condition and provides a reference point to the feedback control system, and subsequently generates control action to devise the traffic towards the reference states to achieve certain system optimal objectives designated by the traffic management authorities. The procedure needs to be conducted cyclically and frequently in a rolling horizon fashion, in contrast to the descriptive evacuation models focusing on the testing of fixed evacuation plans. The concept of the prescriptive models is illustrated in Fig. 3.



Emergency Evacuation, Dynamic Transportation Models, Figure 3 General modeling framework for prescriptive evacuation models

Key Applications

Dynamic transportation planning and operations models and GIS are increasingly applied in the planning of mass evacuation. Most known applications include the development of a decision support system [5,47,48,49,50,51,52] for evacuation planning, evaluation of operational strategies like contra-flow lanes [43,53,54,55], household trip chain sequencing [56], or simulation of traffic flows during evacuation [43,57,58,59].

Future Directions

There is a long history of transportation modeling and GIS application in evacuation management. The role of GIS in such an application has evolved and has been enhanced through the integration with specialized simulation models with the goals of enabling more realistic modeling of evacuation traffic modeling than static approximation of evacuation traffic. However, the majority of such endeavors are still primarily research oriented. More efforts are still needed to better integrate the existing GIS tools with the emerging dynamic modeling approaches. As indicated above, both descriptive and prescriptive dynamic evacuation models need to deal with the time-dependence of the spatial network. To be able to utilize GIS to better understand and analyze network dynamics, graph databases that represent spatial networks need to represent the temporal nature of the transportation network. Therefore, further work is needed to address the time-varying aspect of spatial databases with special focus on supporting emergency evacuation.

Cross References

► Contraflow for Evacuation Traffic Management
► Contraflow in Transportation Network
► Emergency Evacuation Plan Maintenance
► Emergency Evacuations, Transportation Networks

Recommended Reading

1. Urbanik, T., Desrosiers, A.E.: An Analysis of Evacuation Time Estimates around 52 Nuclear Power Plant Sites. US Nuclear Regulatory Commission (1981)
2. Sheffi, Y., Mahmassani, H.S., Powell, W.B.: A Transportation Network Evacuation Model. Transp. Res., Part A **16**, 209–218 (1982)

3. KLD (1094): Formulations of the DYNEV and I-DYNEV Traffic Simulation Models Used in ESF. Federal Emergency Management Agency
4. Lindell, M.K., Bolton, P.A., Perry, R.W., Stoetzel, G.A., Martin, J.B., Flynn, C.B.: Planning Concepts and Decision Criteria for Sheltering and Evacuation in a Nuclear Power Plant Emergency. National Environmental Studies Project, Atomic Industrial Forum, Inc., AIF/NESP-031 (1985)
5. Han, A.F.: TEVACS: Decision Support System for Evacuation Planning in Taiwan. J. Transp. Eng. **116**, 821–830 (1990)
6. United States Nuclear Regulatory Commission: The Accident At Three Mile Island Fact Sheet. United States Nuclear Regulatory Commission, Washington, D.C. (2000)
7. Hobeika, A.G., Radwan, A.E., Jamei, B.: Transportation Actions to Reduce Evacuation Times Under Hurricane/Flood Conditions: A Case Study of Virginia Beach City. Presented at Annual Meeting of Transportation Research Board, Washington, D.C., Jan 1985
8. Ardekani, S.A., Hobeika, A.: Logistics Problems in the Aftermath of the 1985 Mexico City Earthquake. Transp. Q. **42**, 107–124 (1988)
9. Ardekani, S.A.: A Decision Tool for Transportation Operations Following Urban Disasters, Final Report. National Science Foundation, The University of Texas at Arlington, Arlington (1992)
10. Oak Ridge National Laboratory: Oak Ridge Evacuation Modeling System (OREMS) User' Guide. Oak Ridge National Laboratory, Oak Ridge (1999)
11. PBS&J: Southeast United Sates Hurricane Evacuation Traffic Study: Evacuation Travel Demand Forecasting System, Technical Memorandum 2, Final Report. PBS&J, Tallahassee (2000)
12. PBS&J: Reverse Lane Standards and Its Strategies. PBS&J, Southeast United States Hurricane Study, Tallahassee (2000)
13. Lindell, M.K., Prater, C.S.: Development of a decision support system for evacuation management. Presented at the First International Conference on Debris-Flow Disaster Mitigation Strategy, Taipei, Taiwan, 2002
14. Burton, M.L., Hicks, M.J.: Hurricane Katrina: Preliminary Estimates of Commercial and Public Sector Damages. Center for Business and Economic Research, Marshall University, Huntington (2005)
15. Alsnih, R., Stopher, P.R.: Review of Procedures Associated With Devising Emergency Evacuation Plans. Transportation Research Record. pp. 89–97. Transportation Research Board, Washington, D.C. (2004)
16. U.S. Department of Homeland Security: Target Capabilities List: Version 1.1. U.S. Department of Homeland Security, Office of State and Local Government Coordination and Preparedness, Washington, D.C. (2005)
17. U.S. Department of Homeland Security: National Incident Management System. U.S. Department of Homeland Security, Washington, D.C. (2004)
18. London Resilience Team: London Prepared (2005)
19. Spanish Minister of Interior: Framework for Action 2005–2015 (2005)
20. Farradyne, P.B.: A Guide to Updating Highway Emergency Response Plans for Terrorist Incidents. The American Association of State Highway and Transportation Officials' Security Task Force (2002)
21. Rowshan, S.: A Guide to Highway Vulnerability Assessment for Critical Asset Identification and Protection. FHWA, pp. 47 (2002)
22. Fink, C.N.: Anti-Terrorism Security and Surface Transportation Systems: A Review of Case Studies and Current Tactics. Presented at Annual Meeting of Transportation Research Board, Washington, D.C., Jan 2003
23. Rowshan, S., Smith, M.C., Krill, S., Seplow, J.E., Sauntry, W.C.: A State DOT Guide to Highway Vulnerability Assessment, AASHTO (2003)
24. Lewis, D.C.: September's Great Escape: New Information System Helps Manage Hurricane Evacuations. Roads Bridges **September**, 40–42 (2001)
25. Pal, A., Graettinger, A.J., Triche, M.H.: Emergency Evacuation Modeling Based On Geographical Information System Data. Transportation Research Board (CD-ROM) 82nd Annual Meeting, Washington, D.C. (2003)
26. Mitchell, J., Reece, C.F.: Leveraging GIS to Manage Weather-Related Evacuations. Presented at The 14th Annual Meeting of ITS America, San Antonio, Oct 2004
27. PBS&J: Floyd Behavioral Reports (Evacuation Traffic Information System), vol. 2005. PBS&J, Tallahassee, Florida (2005)
28. Hobeika, A.G., Jamei, B.: MASSVAC: a Model for Calculating Evacuation Times under Natural Disasters. Presented at Computer Simulation in Emergency Planning. Society for Computer Simulation, San Diego (1985)
29. ORNL: Oak Ridge Evacuation Modeling System (OREMS) User' Guide. Oak Ridge National Laboratory, Oak Ridge (1999)
30. Mahmassani, H.S., Hayssam, A.S., Zhou, X.: DYNASMART-P V1.2 User's Guide. University of Maryland at College Park, Austin (2006)
31. Ben-Akiva, M., Bierlaire, M., Koutsopoulos, H., Mishalani, R.: DynaMIT: a simulation-based system for traffic prediction. Presented at DACCORD Short Term Forecasting Workshop, Delft (1998)
32. Chiu, Y.C., Zheng, H., Villalobos, J., Gautam, B.: Modeling No-Notice Mass Evacuation Using a Dynamic Traffic Flow Optimization Mode. IIE Transactions **39**, 83–94 (2007)
33. Liu, Y., Lai, X., Chang, G.L.: A Two-Level Integrated Optimization Model for Planning of Emergency Evacuation: A Case Study of Ocean City Under Hurricane Evacuation. Presented at The 84th Annual Meeting of Transportation Research Board, Washington, D. C. (2005)
34. Chiu, Y.C., Zheng, H.: Real-Time Mobilization Decisions for Multi-Priority Emergency Response Resources and Evacuation Flows: Model Formulation and Solution. Transportation Research Part E (in press 2007)
35. Sherali, H.D., Carter, T.B., Hobeika, A.G.: A location-allocation model and algorithm for evacuation planning under hurricane/flood conditions. Transp. Res. Part B, 439–452 (1991)
36. Hagani, A., Oh, S.C.: Formulation and Solution of a Multi-Commodity, Multi-Modal Network Flow Model for Disaster Relief Operations. Transp. Res. Part A **30**, 231–250 (1996)
37. Brown, G.G., Valliliou, A.L.: Optimizing Disaster Relief: Real-Time Operational and Tactical Decision Support. Naval Res. Logistics **40**, 1–23 (1993)
38. KLD: Development of IDYNEV and PCDYNEV. http://www.kldasociates.com/evac.htm (2004)
39. Kwon, E., Pitt, S.: Evaluation Of Emergency Evacuation Strategies For Downtown Event Traffic Using a Dynamic Network Model. Presented at Annual Meeting Of Transportation Research Board, Washington, D.C. (2005)
40. Henk, R.H., Ballard, A.J., Robideau, R.L., Peacock, W.G., Maghelal, P., Lindell, M.K., Prater, C.S., Loftus-Otway, L.,

Siegesmund, P., Harrison, R., Lasdon, L., Chiu, Y.C., Zheng, H., Perkins, J., Lewis, C., Boxill, S.: Disaster Preparedness in Texas. College Station, Texas Transportation Institute (2007)
41. Ben-Akvia, M., Bierlaire, M., Koutsopoulos, H., Mishalani, R.: DynaMIT: a simulation-based system for traffic prediction. In: DACCORD Short Term Forecasting Workshop. Delft, Netherlands (1998)
42. FHWA: TSIS Manuals and Documentation. FHWA, Washington, D.C. (2003)
43. Chiu, Y.C., Villlalobos, J.A., Mirchandani, P.B.: The Development of the Multi-Resolution Assignment and Loading of Traffic Activities (MALTA) for Regional Mass Evacuation Analysis in Central Texas, (2007)
44. Chiu, Y.C., Zhang, H., Villalobos, J.A., Gautam, B.: Modeling no-notice mass evacuation using a dynamic traffic flow optimization model. IIE Trans **39**, 83–94 (2007)
45. Chiu, Y.C., Korada, P., Mirchandani, P.B.: Dynamic Traffic Management for Evacuation. Presented at The 84th Annual Meeting of Transportation Research Board, Washington, DC (2004)
46. Liu, H., Ban, J., Ma, W., Mirchandani, P.B.: Adaptive Control Based Real Time Traffic Management for Emergency Evacuation. ASCE J. Urban Plann. Dev. (in press 2007)
47. Tufekci, S., Kisko, T.M.: Regional Evacuation Modeling System (REMA): A decision support system for emergency area evacuation. Computers & Industrial Engineering. Elsevier (1991)
48. Tufekci, S.: An integrated emergency management decision support system for hurricane emergencies. University of Florida, Safety Science, July 1995 (1995)
49. Nisha de Silva, F.: Challenges in Designing Spatial Decision Support Systems for Evacuation Planning. Natural Hazards Research Working Paper, (2000)
50. Hobeika, A.G., Ardekani, S.A., Martinez-Marquez, A.: A Decision Support System for Emergency Transportation Operations Following Major Urban Earthquake: a Case Study of the 1985 Mexico City Earthquake. National Science Foundation ECE-8610841 (1987)
51. Lindell, M.K., Prater, C.S.: Development of a decision support system for hurricane evacuation management, pp. 238–294. Iowa State University, Armes, Iowa (2002)
52. Nisha de Silva, F.: Challenges in Designing Spatial Decision Support Systems for Evacuation Planning. Natural Hazards Research Working Paper, (2000)
53. Theodoulou, G., Wolshon, B.: Modeling and Analyses of Freeway Contraflow to Improve Future Evacuations. Presented at The 83rd Annual Meeting of Transportation Research Board, Washington, D.C. 2004
54. Kwon, E., Pitt, S.: Evaluation of Emergency Evacuation Strategies for Downtown Event Traffic Using a Dynamic Network Model. Transp. Res. Record: J. Transp. Res. Board **1922**, 149–155 (2005)
55. Lambert, L., Wolshon, B.: Analysis of Reverse Flow Traffic Operations, Phase I: Urban Sporting Event Measurement and Evaluation. Louisiana State University, Department of Civil and Environmental Engineering, Baton Rouge, LA (2002)
56. Murray, P.M., Mahmassani, H.S.: Model of Household Trip Chain Sequencing in an Emergency Evacuation. Transportation Research Board (CD-ROM), 82nd Annual Meeting, Washington, D.C (2003)
57. Rathi, A.K., Solanki, R.S.: Simulation of traffic flow during emergency evacuation. Presented at Simulation Conference, 1993
58. Han, L.D., Franzese, O.: Modeling Mass-Evacuation Traffic Operations Due to Hurricanes. Presented at ITE 2001 Annual Meeting and Exhibit, Chicago, Illinois, 2001
59. Sinuany-Stern, Z., Stern, E.: Simulating the evacuation of a small city: the effects of traffic factors. Socio-Economic Planning Sciences, June 1993. Elsevier (1993)

Emergency Evacuation Models

► Emergency Evacuation, Dynamic Transportation Models

Emergency Evacuation Plan Maintenance

CHENG LIU, MARK TUTTLE
Geographic Information Science & Technology Group, Computational Science & Engineering,
Oak Ridge National Laboratory, Oak Ridge, TN, USA

Synonyms

Contingency management system; Incident management system; Evacuation logistics planning; Massive evacuations; Emergency preparedness & response

Definition

Emergency evacuation is the process of moving people out of a dangerous area as quickly and orderly as possible. The threats could be precipitated by natural or man-made events. For example, an accidental fire, a radiation emission, a chemical explosion, a bomb threat, or a major hurricane. The goal of a carefully designed emergency evacuation plan is to optimize the usage of available transportation resources and reduce the evacuation time in order to save lives when the disaster occurs.

The evacuation area can be as small as a room or a building, or as large as multi-county metropolitan area. The simplest evacuation plan could be the exit signs directing people to the nearest egress point combined with a public announcement system inside the building. For a mass evacuation of large areas for events such as hurricanes or floods, the evacuation planning process is more complex. It typically starts with the determination of the risk area; that is, the zone where the population is under risk. If emergency evacuation is required, it is necessary to estimate how long it will take to evacuate all of the population inside the area and how many resources are needed to evacuate this area. Evacuation logistics include the fuel required for all vehicles, the public transit of disabled population, emergency response personal and emergency medical equipment.

Evacuation types include mandatory, recommended, and voluntary. Even if the evacuation is mandatory, there are still people who remain in the danger area for a variety of reasons: to protect property, afraid of the traffic jams, and limited personal resources during the evacuation. In this circumstance, public officials and emergency response personnel must be able to respond to the needs of these people. All of these key features should be carefully considered in the emergency evacuation planning process.

Historical Background

The Federal Emergency Management Agency (FEMA) is responsible for developing plans and assisting with recovery from disasters. FEMA can trace its beginnings to the Congressional Act of 1803. This act, the first piece of disaster legislation, provided assistance to a New Hampshire town following an extensive fire. In 1979, President Carter's executive order merged more than 100 federal agencies which were disaster-related into a new Federal Emergency Management Agency. In March 2003, it became part of the new Department of Homeland Security [1].

The first on the scene during an emergency evacuation is the local public safety and emergency response agencies. Similar to the US government, each state has its own history related to regulatory authority related disaster preparedness and emergency response. In general, the major components of the State command structure include: Governor, Chief Executive Officer, State Emergency Operations Center, State Coordinating Officer, Unified Area Command Structure, State Emergency Management Agency, Local Chief Executive Officer, and Law Enforcement Agencies [3].

Scientific Fundamentals

If the evacuation area is a building or a complex of buildings such as an industrial plant, the emergency evacuation planning is simple. Thus, the evacuation plan can be easily validated by executing a few real time exercises. However, in a mass evacuation of a large area, the planning process is complex and extremely expensive, and it is therefore, in most cases, unfeasible to validate the plan by executing a real time evacuation exercise. The creation and validation of most emergency evacuation plans for mass evacuation are based on complex computer models.

The ability of a Geographic Information System (GIS) for combining spatial data into one system makes it an essential tool for emergency evacuation planning. For example, hurricane and air diffusion models can estimate the effected area or contaminant plumes and hence determine the area at risk or area requiring evacuation. A demographic model can estimate the populations which are in the risk area and their locations and social characteristics such as age, gender, and income. Transportation models estimate the evacuation time, identify congested areas and aid in the design of plans to divert/control the traffic in order to reduce the traffic and decrease the evacuation time.

Many computer models are developed for specific disasters that will aid the evacuation planning process. For example, the Sea, Lake and Overland Surges from Hurricanes (SLOSH) was developed by the National Weather Service to predict hurricane storm surges for a given set of conditions. It is also used to plan evacuation routes and locate emergency shelters based on estimates of which geographic areas could be flooded in certain storm scenarios [2]. BOSS DamBrk is used for dynamic flood routing, dam safety analysis, and reservoir spillway analysis and was developed by the FERC, US Army Corps of Engineers, state regulatory agencies and consultants worldwide [DAMBRK]. HURREican EVACuation (HURREVAC) tracks and estimates the course of an approaching hurricane and projects the course once it makes landfall. HURREVAC was developed by the USACE for FEMA. Multi-Hazards US Software (HAZUS-MH) predicts the disasters with regards to buildings and structures from various hazards and was developed by FEMA. The Consequence Assessment Tool Set (CATS)/Joint Assessment of Catastrophic Events (JACE) integrates an array of information to provide disaster analysis in real time and was developed by the Defense Treat Reduction Agency for FEMA [2].

Traffic analysis models are also important in emergency evacuation planning. These models estimate the evacuation time and identify the congested areas. Based on the size of the evacuation area and required computational resources for execution of the model, the traffic analysis models can be categorized into three groups: macroscopic, mesoscopic, and microscopic models.

Macroscopic traffic models are aggregation models which are based on traffic equilibrium (static or dynamically) properties. These models treat each link as an individual entity and can cover a larger area within a reasonable execution time. Another macroscopic model is based on system optimization using network flow and routing algorithms. These models formulate the evacuation process as a linear programming (LP) or a mixed linear programming (MIP) problem. The objective is to minimize the evacuation time or maximize the traffic flows subject to the network infrastructure constraints [3].

Microscopic models attempt to simulate and analyze individual travelers (vehicles) during an evacuation. Those models include the details of driver behavior (car following, lane switching, etc.), signal control, and behavioral dynamics during an evacuation event. Microscopic models require more detailed geospatial data such as population distribution and demographic information. These models generate more realistic results. Unfortunately, when the evacuation area becomes large, computer processing resources become significant. Typically, microscopic evacuation models require the resources of high performance computers (HPC) to solve the models for a large area.

The mesoscopic models preserve driver behavioral characteristics such as car following, gap acceptance and lane changing but sacrifice the detailed location of each vehicle on the link like the microscopic simulation model. The goal of these models is to capture the temporal congestion phenomena with faster evacuation time and less detailed vehicle location.

Software for Emergency Evacuation Planning

There are many transportation simulation software applications that can be used for emergency evacuation planning which are available from private companies, universities, national laboratories or departments of transportation. Such examples include the NETwork emergency eVACuation (NETVAC), MASS eVACuation (MASSVAC), Oak Ridge Evacuation Modeling Systems (OREMS) [4]. Other transportation simulation software that can be used for emergency evacuation with slight modifications are Paramics [5], VISSIM [6], CORridor microscopic FLOw simulation (CORFLO), INTEGRATION, CORridor microscopic SIMulation (CORSIM), MIcroscopic Traffic SIMulation Laboratory (MITSMILab) [7], DynaMIT, DYNASMART [8], and the Transportation Analysis SIMulation System (TRANSIMS) [9]. It is worth noting that MITSIMLab and TRANSIMS are open-source projects and the source code can be downloaded from the internet. A few examples are listed below.

OREMS

OREMS is a Window-based software program designed to analyze and evaluate large-scale vehicular emergency evacuations, conduct evacuation time estimation studies, and develop evacuation plans [4].

MITSIMLab

MITSIMLab is a simulation-based laboratory that was developed for evaluating the impacts of alternative traffic management system designs at the operational level and assisting in subsequent refinement [7]. It has three components, Microscopic Traffic Simulator (MITSIM), Traffic Management Simulator (TMS) and Graphic User Interface (GUI).

TRANSIMS

TRANSIMS is an agent-based simulation system capable of simulating the second-by-second movements of every person and every vehicle through the transportation network of a large metropolitan. It offers a set of multimodal transportation software options for travel demand and microscopic transportation simulation. These models are very capable of simultaneously modeling pedestrian, biking, vehicle and transit transportation [9]. The open source version does not have a graphic user interface. However, a commercial GUI (TRANSIMS-Visualizer) is available from IBM.

E

Geospatial Data for Evacuation Modeling

The fundamental data for emergency evacuation planning is transportation network, traffic controllers and population distribution and characteristics. The US Census Bureau is the most widely available source for population and household economic information for the USA. LandScan offers the population distribution in a cell-based format. Tiger/Line files, Tele Atlas and NAVTEQ are examples of nation-wide transportation network data sets. There is currently no national database for traffic controllers at this stage. A few examples are listed in the following text.

Census Population and Housing Datasets

The Census Bureau is the leading source of quality data about the nation's people and socio-demographic characteristics. It provides population distribution information on the block level for metropolitan areas. This information may be used as the basic population information for the emergency evacuation plan.

Landscan Population Datasets

LandScan Global is global population distribution datasets which have a grid cell size of 30 arc seconds (approximately 1 km × 1 km at the equator) and is appropriate for very large area evacuation modeling. LandScan USA is a finer resolution (90 m × 90 m) population distribution data for United States and includes both an average daytime population distribution and a residential (nighttime) population distribution. Both datasets are developed at the Oak Ridge National Laboratory [10].

Topologically Integrated Geographic Encoding and Referencing System (TIGER)

TIGER polygon files contain information about the boundaries of Census enumeration units such as block, block group, zip codes, census tract, and traffic analysis

Emergency Evacuation Plan Maintenance, Figure 1 West Knoxville evacuation area

zones.TIGER Line layers contain street information which is important for transportation simulation models.

NAVTEQ

Navteq is a GIS data provider providing detailed road network including up to 204 attributes such as turn restrictions, physical barriers and gates, one-way streets, restricted access points and relative road heights.

Tele Atlas

Tele Atlas's Dynamap/2000 data is another data source for road network information.

Data Example

Figure 1 shows the Tele Atlas street data and LandScan USA cell data for a portion of West Knox County, TN. The square dots represent the centers of the 90 meter LandScan USA cells. The circle represents the evacuation area.

Key Applications

An evacuation plan is not only a written document but a planning process. A mass emergency evacuation plan should be coordinated and integrated with plans from federal, state and local agencies. A recent study by the US Nuclear Regulatory Commission (NRC) estimated that there were 14 evacuations of more than 100,000 people between 1990 and 2003. Of these large-scale evacuations, there were two evacuations which involved more than one million people. Although the tools discussed previously have been used for emergency evacuation planning, there is no single tool that meets all requirements [3]. Easy to use and comprehensive emergency evacuation planning tools are in demand.

Furture Directions

Emergency evacuation planning includes many complicated tasks. Two of the current study concentrations are evacuee behavior and evacuation logistics. Evacuee behaviors are different under different types of evacuations. In a scheduled emergency evacuation, each evacuee has enough time to plan their evacuation route, destination, departure time and returning time. Under non-scheduled mandatory emergency evacuation execution, clearing the evacuation area as quickly as possible remains the goal; but evacuees may also need to pick up school kids prior to departing the at-risk area or address refueling issues while departing the evacuated area. This results in a significantly more complex evacuation plan and modeling these behavioral characteristics is a challenge. Furthermore, during a non-scheduled voluntary evacuation, public officials must have methodologies in place to locate and monitor the population who choose to remain in the risk area. The evacuation plan must also provide for contingencies should the voluntary evacuation become a mandatory evacuation. Large scale evacuation models and simulations should also includes logistics such as adequate fuel supply for all evacuees and provision of fuel to evacuee's during the event and disruption of traffic controls.

Cross References

► Nearest Neighbors Problem

Recommended Reading

1. FEMA: FEMA History. http://www.fema.gov/about/history.shtm. Accessed 10 Sept 2007
2. US Department of Transportation, US Department of Homeland Security: Report to Congress on Catastrophic Hurricane Evacuation Plan Evaluation, June 1 (2006)
3. Lu, Q., George, B., Shekhar, S.: Capacity Constrained Routing Algorithms for Evacuation Planning: A Summary of Results, Advances in Spatial and Temporal Databases. In: Proceeding of 9th International Symposium on Spatial and Temporal Databases (SSTD'05), Angra dos Reis, Brazil, 22–24 Aug 2005
4. Franzese: Oak Ridge Evacuation Modeling System (OREMS), User's Guide. Oak Ridge National Laboratory. Oak Ridge, Tennessee (2005)
5. Quadstone Paramics: http://paramics-online.com/index.php. Accessed 10 Sept 2007
6. Road Traffic Technology: http://www.roadtraffictechnology.com/contractors/traffic_man/ptv_ag/ptv_ag2.html. Accessed 10 Sept 2007

7. Mahmassani, H.S., Sbayti, H., Zhou, X.: DYNASMART-P, Intelligent Transportation Network Planning Tool, Version 1.0 User's Guide, Unversity of Maryland, August (2004)
8. Ben-Akiva, M.: User's Guide for MITSIMLab and Road Network Editor (RNE), Massachusetts Institute of Technology, Intelligent Transportation Systems Program, November (2002)
9. Los Alamos National Laboratory: http://transims.tsasa.lanl.gov/. Accessed 10 Sept 2007
10. Budhendra, B., Bright, E., Coleman, P., Dobson, J.: LandScan: Locating People is What Matters. Geoinfomatics **5**(2), 34–37 (2002)

Emergency Evacuations, Transportation Networks

OSCAR FRANZESE[1], CHENG LIU[2]
[1] Engineering Science and Technology Division, Oak Ridge National Laboratory, Oak Ridge, USA
[2] Geographic Information Science & Technology Group, Oak Ridge National Laboratory, Oak Ridge, TN, USA

Synonyms

Roadway network model; Evacuation model systems; Consequence management; Immediate response zone; Protective action zone

Definition

Evacuation modeling systems (EMS) have been developed to facilitate the planning, analysis, and deployment of emergency evacuation of populations at risk. For any EMS, data such as road network maps, traffic control characteristics, and population distribution play critical roles in delineating emergency zones, estimating population at risk, and determining evacuation routes.

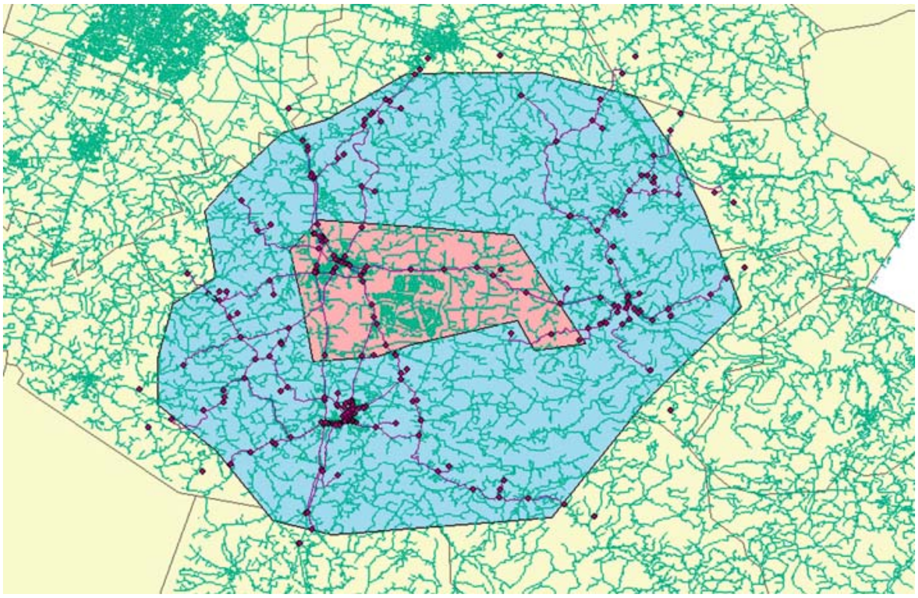
There are situations in which it is possible to plan in advance for an emergency evacuation including, for example, an explosion at a chemical processing facility or a radiological accident at a nuclear plant. In these cases, if an accident or a terrorist attack were to happen, then the best evacuation plan for the prevailing network and weather conditions would be deployed. In other instances—for example, the derailment of a train transporting hazardous materials—, there may not be any previously developed plan to be implemented and decisions must be made adhoc on if and how to identify and proceed with the best course of action to minimize losses. Although both cases require as a starting point the development of a transportation network model of the area at risk, which must include road capacity and topology, in the latter the available time to generate this network is extremely limited. This time constraint precludes the use of any traditional data gathering methodology and the network generation process has to rely on the use of GIS and stochastic modeling techniques. The generation of these transportation networks in real time is the focus of this entry.

Historical Background

The 1995 Presidential Decision Directive (PDD) 39 introduced the nation to the term *Consequence Management* which meant to establish how the effects of a terrorist attack employing weapons of mass destruction and/or other means would be managed. In contrast to *Crisis Response*, defined in the PDD-39 as actions to be taken when a potential attack has been discovered and before any damage could be inflicted to the infrastructure and/or the population, *Consequence Management* encompasses those measures aimed at alleviating the physical, socioeconomic, and psychological effects of a biological, chemical or other type of attack. Those measures include any preliminary actions to be taken in preparation for a potential crisis, as well as any strategy to be deployed if the incident develops. For example, in the case of a sport event where large attendance is expected, the preliminary actions may include surveys of the venue and its surroundings, determination of the availability and ability of local and regional hospitals to treat and/or decontaminate victims, and the assessment of the location and condition of various antidotes. The post-incident actions, in the same example, may include plans for the treatment of the victims, their decontamination (if necessary), evacuation, and cleanup.

The importance of Consequence Management became more apparent after 9/11 as it was realized that is impossible to harden the infrastructure to avoid losses in a range of damage scenarios. If damage occurs, the goal is to minimize those loses by an effective response and to achieve a rapid recovery. The nation's new awareness of the need for enhanced security measures made a compelling case for establishing a consequence management simulation capability on a scale that is commensurate with national needs, which could analyze different course of actions, predict their consequences, and provide information to decision-support systems [1].

One of these potential courses of actions is evacuation of the area affected. However, before the evacuation alternative can be evaluated, the assessment analysis major challenge is the impromptu generation of all the inputs for the traffic simulation model, including the delineation of the potentially affected area, the population spatial distribution at the time of the event, and the topology and capacity of the transportation network within the area at risk and beyond. This is not an easy task since the main characteristic of these events is that they can happen anywhere anytime, and provide a very short response time.



Emergency Evacuations, Transportation Networks, Figure 1 Network Topology within the IRZ and PAZ areas

Scientific Fundamentals

For events with limited or non-existent advance notice, the analyst would mark on a digital map the point at which the event occurred. With information about the type of agent that has been released and the topography of the area, the boundaries of the Immediate Response Zone (IRZ, or the zone where an effective and prompt response is critical in order to avoid the loss of human lives) and the Protective Action Zone (PAZ, or the area beyond which no adverse effects may be expected for the population) are then delineated with models that predict how the toxic gases or other noxious substances would mix and interact with air and the distance that they may travel until becoming innocuous. Since the boundaries of the PAZ are such that when it is crossed the evacuating population is no longer at risk, then the PAZ delimits the transportation network that needs to be considered for the traffic simulation model of the evacuation.

Once the IRZ and PAZ have been established, a network creation module (NCM) extracts the topology of the transportation network by using spatial information databases ranging from simple and publicly available (i. e., TIGER files) to very sophisticated. Several steps follow which consist on building the intersections between links, creating nodes representing these intersections, generating "from" and "to" node information for each link, eliminating duplicated links, and extracting the highway network with information about the road class. After that, the data is checked for inconsistencies (e. g., misclassified road attributes) and road connectivity (e. g., adding links where missing but not where are physical barriers such as rivers; adding freeway ramps if missing). To complete the topology of the transportation network, exit nodes are added, i. e., nodes at the edge of the PAZ. Figure 1 shows the IRZ (central polygon) and the PAZ (outer polygon) for a 2,000 sq mile analysis area. The figure shows the basic transportation network and the network extracted by the model presented here (darker, with small diamonds representing nodes).

At this point, the transportation network is not complete, since traffic models, particularly traffic simulation models, need not only topology information but also road and intersection capacity. Depending on the spatial database used in the process, road capacity (e. g., number of lanes per link) may be available. If this information is not readily available, then probabilistic models that assign attributes to segments of roadways based on their classification and location need to be built.

One way of developing these probabilistic models is by using publicly available information such as the FHWA Highway Performance Monitoring System, (HPMS) database. This is a national level highway database that includes use, performance, and operating characteristics information on all public roads in the US. Using the HPMS database, probability distributions of traffic parameters—such as number of lanes, speed limit, spatial location of the roadway facility (urban/rural), type of traffic control, and other characteristics, can be built for each State and each roadway class within that State. These probability distributions, in turn, provide the inputs for Monte Carlo simulation techniques that are used to generate transportation network attributes and parameters that are not contained in other spatial databases [2].

These probabilistic assignments of traffic parameters cannot be made in isolation for each segment of roadway within the generated transportation network, since each one of these segment has an upstream and a downstream roadway segment that are very likely to have the same parameters [3,4]. For example, within an urban area a segment of freeway f between exits i and $i+1$ is very likely to have the same number of lanes as the upstream segment $f-1$ between exits $i-1$ and i. That is, the probability distributions not only need to be built as conditional probability distributions, but once the parameters have been assigned they need to be checked using GIS techniques that permit to verify that these parameters are changing in a consistent manner. In the previous example, if the probabilistic model assigned four lanes to segments $f-2, f-1, f+1$, and $f+2$, and two to segment f, then it would be reasonable to override the probabilistic model based on the topology of the network and assign four lanes to segment f as well.

Once the basic roadway capacity (i. e., number of lanes and speed limit) has been determined, intersection capacity (i. e., traffic signal controllers or signs) can be generated based on the same probabilistic models, plus information on the class and (already generated) capacity of the two links intersecting. The determination of the type of traffic control at the intersection, as well as traffic signal control settings (if that is the type of control assigned to that intersection) is a more difficult problem that requires the combination of the information provided by the GIS engine and traffic signal/sign design methods. Key variables such as the number of lanes on each of the two intersecting streets, their location (urban/rural), and the roadway class of each of these two streets determines if the intersection is more likely to have a traffic signal than a sign or not. For example, an urban intersection between two arterials with four lanes each has an almost 100% probability of having a traffic signal controller, while a rural intersection between two two-lane streets is very likely to have signs or no control at all. If a traffic signal is assigned (first case in the previous example), then the same variables are used to design the traffic control parameters on the fly.

Key Applications

Transportation Network Models are used mainly in the areas of Transportation Planning and Traffic Operations. Other applications may include training, especially in the area of Emergency Management.

Transportation Planning

Traditionally, the data used to support EMS and other transportation related applications is collected indirectly from different sources, including population census, existing databases, field data, and economic statistics. Although this traditional methodology can provide reasonably accurate information, it is a lengthy process which cannot be used when time is of the essence such in the cases of no-advance-notice evacuation modeling and evaluation. In those cases, a methodology such as the one presented here is more adequate, even though it may not be as accurate as a traditional data gathering method. Similarly, this methodology is also applicable to other transportation modeling applications where high accuracy is not an overarching concern but data collection cost is important.

Traffic Operations

In general, transportation operations requires accurate data, therefore the methodology presented here may not be appropriate for these applications. However, in cases where no data is available and data collection time is a key factor in helping decide the feasibility of a given course of action (e. g., evacuation of an area at risk), then if that alternative is selected its implementation (traffic operations) will have to be accomplished using the information at hand. This information can be enhanced by using other GIS/Remote Sensing technologies to refine the data at key places in the transportation network.

Emergency Management

Emergency management agencies frequently organize drills aimed at handling localized incidents which, although very valid for many emergency functions, do not provide the experience necessary to deal with more extensive traffic management strategies required in scenarios such as those described before. One of the main reasons for this is that drills or exercises involving public participation and covering large geographic areas are not feasible. Those drills can only be played using traffic simulation software which permits to recreate real-world situations and generate real-time feedback. The procedures described here provide a very cost effective data gathering method for those traffic simulation models.

Future Directions

The main area of future research focuses on the accuracy of the transportation network topology and roadway attributes that can be attained using the methods presented here. First, it is first necessary to determine how different these traffic network parameters generated using the methodologies described here are from traffic parameters obtained using more traditional methodologies. Some discrepancies between the two are expected. If these discrepancies are large, then the accuracy of the transportation network topology and roadway attributes that the methods described here would need to be enhanced. The network topology can be improved by using more sophisticated GIS databases as the starting point for the methods described here. More problematic is to increase the roadway and intersection capacity accuracy. For selected and critical points on the network (which can be identified after one pass of the traffic simulation models), this accuracy can be improved through the use of GIS and Remote Sensing technologies which are currently under research. That is, after the simulation model has run with the traffic parameters obtained during the first pass of the models described here, then critical intersections (i. e., those that are expected to play major roles during the evacuation) can be identified. Remote sending techniques combined with image analysis and automatic feature extraction can then be used to refine the main parameters at these critical intersections.

Future expansions of the methodology will focus on the integration of the models presented here with plume dispersion models. The latter play a key role not only in determining the areas that may require evacuation, but also on which parts of the transportation networks would be available during an evacuation and hence need to be modeled. Besides the nature of the compound of the element released to the atmosphere, these plume dispersion models also require information regarding the topography of the area, the land cover, as well as real-time weather patterns.

Cross References

► Emergency Evacuation, Dynamic Transportation Models
► Road Network Data Model
► Weather Information, Surface Transportation

Recommended Reading

1. National Research Council: Making the Nation Safer: The Role of Science and Technology in Countering Terrorism. National Academy Press, Washington DC (2002)
2. Franzese, O., Sorensen, J.: Fast Deployable System for Consequence Management: The Emergency Evacuation Component. In: Proceedings of the 2004 ITS Safety and Security Conference, Miami Beach, FL (2004)
3. Liu C., Franzese, O., Bhaduri, B.: Oak Ridge Evacuation Modeling System (OREMS): A PC-based Computer Tool for Emergency Evacuation Planning. In: Proceedings of the GIS for Transportation Symposium, Columbus, OH (2006)
4. Liu C., Franzese, O., Bhaduri, B.: A Front-end Geographic Information System for an Emergency Evacuation Model. In: Proceedings of the Association of American Geographers, Chicago, IL (2006)

Emergency Preparedness

► Contraflow for Evacuation Traffic Management

Emergency Preparedness & Response

► Emergency Evacuation Plan Maintenance

Emergency Response

► Contraflow in Transportation Network

Energy-Aware

► Data Collection, Reliable Real-Time

Energy Constraint

► Geosensor Networks, Estimating Continuous Phenonena

Energy Optimization

► Data Collection, Reliable Real-Time

Entity Integration

► Conflation of Features

Environmental Communication

► Wayfinding: Affordances and Agent Simulation

Environmental Criminology

► Crime Mapping and Analysis

Environmental Modeling Using Open Source Tools

ARI JOLMA[1], DANIEL P. AMES[2], NED HORNING[3], HELENA MITASOVA[4], MARKUS NETELER[5], AARON RACICOT[6], TIM SUTTON[7]
[1] Helsinki University of Technology, Espoo, Finland
[2] Department of Geosciences, Geospatial Software Lab, Idaho State University, Pocatello, ID, USA
[3] Center for Biodiversity and Conservation, American Museum of Natural History, New York, NY, USA
[4] Department of Marine, Earth and Atmospheric Sciences (MEAS), North Carolina State University, Raleigh, NC, USA
[5] Center for Scientific and Technologic Research, The Trentino Cultural Institute, Trento, Italy
[6] Ecotrust, Portland, OR, USA
[7] Reference Center on Environmental Information, Campinas, Brazil

Synonyms

Open-source GIS

Definition

Geospatial data can include socioeconomic, environmental, geophysical, and technical data about the Earth and societal infrastructure and hence is pivotal in environmental modeling and management (EMM). Desktop, web-based, and embedded geospatial systems have become an essential part of EMM, providing pre- or post-processing of geospatial data, analysis and visualization of results using a graphical user interface (GUI). When combined with the growing culture of free and open source software (FOSS), this combination of geographic information systems) GIS and environmental modeling can result in powerful and freely accessible tools for environmental modeling and management. This study provides an overview of GIS FOSS software (FOSS4G) as applied to EMM and analyzes platforms, software stacks, and EMM workflows. In the FOSS world, the barriers to interoperability are low and hence FOSS4G tools can significantly enhance collaboration between EMM professionals beyond what is possible when strictly proprietary EMM tools are used. For example, as a FOSS4G-based tool, the US EPA BASINS watershed modeling system source code can be explored, modified, updated, and improved by anyone, ultimately resulting in better EMM systems.

Historical Background

The growing culture of FOSS has produced a new breed of software and associated business models. The community that develops FOSS4G has recently gained momentum [1], making it worthwhile to study the applicability and current applications of FOSS4G in EMM. The development and use of environmental models with geospatial data and tools, including FOSS4G, has been studied for several decades. The initial phase of this research culminated in a series of conferences such as the first HydroGIS [2] in 1996 and the first GIS/EM [3] in 1992, which presented the then state-of-the-art in development and application of geospatial tools in EMM. For example Mitasova et al. [4] and Frysinger et al. [5] used FOSS and explicitly addressed "openness" at the first GIS/EM conference. FOSS4G continues to grow in acceptance in the GIS community, in part through conferences such as FOSS4G 2006 in Lausanne, Switzerland, with over 500 FOSS4G professionals and developers. Though not focused specifically on EMM, the FOSS4G conferences tend to have a significant number of EMM related papers and presentations and have become the primary venue for presentation of new developments in this field. Many useful lists of FOSS4G tools exist on the internet including the following:

- http://www.freegis.org
- http://www.opensourcegis.org/
- http://giscorps.org/info_for_volunteers/about_free_software.php#Annotated_List_of_Free_and_Open

Scientific Fundamentals

Following is a brief examination of the current state of FOSS4G in EMM with focus on tools that give modelers and managers the freedom of adopting, using, learning, and extending and hence can be considered FOSS [6]. This reflects current understanding of what are or will be the key FOSS building blocks of geospatially aware information systems for environmental problem solving. A systematic look at FOSS4G is focused on the following three main elements:

- Platform: A platform, or a set of platforms, is the medium used to deliver a solution and is a versatile and useful concept for analyzing GIS solutions.
- Software stack: each platform has its own specific needs and tools available for building a working geospatial software stack.
- Workflow: each platform and software stack is particularly suited for specific workflows.

FOSS4G Platforms for Environmental Modeling

Software platforms guide usage of software, may dictate organizational purchases, and direct developers interests. Platforms are defined by operating system, client–server model, and programming environment. Although several GIS programmable environments have existed for many years, these have typically failed to become as popular

Environmental Modeling Using Open Source Tools, Table 1 Desktop free and open source software (FOSS) geospatial software stack based on FOSS tools. *GUI* Graphical user interface, *GIS* geographic information system, *OS* operating system

Generic stack	Geospatial software stack	Grouping
Application extensions/plug-ins, dedicated applications GUI Application	openModeller, SME, Quantum GIS, GRASS, OSSIM, MapWindow GIS	Environmental modeling and data analysis tools. Desktop GIS
Application development tools	Qt, MapWinGIS	User interface and visualization tools
Command line applications	GDAL applications, PostGIS, GMT, GRASS	Data processing, management, and integration
Scripting languages	Perl, Python, Ruby, R,	
Libraries	GeoTools, PostGIS, Proj4, GDAL/OGR, JTS/GEOS	
Programming languages	C, C++, Java, C#, FORTRAN	System software
System libraries	Data exchange libraries, windowing system	
OS kernel	Linux, Mac OSX[1], MS-Windows[1]	

[1]Though Mac OSX and MS-Windows are not FOSS, much of the FOSS geospatial software stack has been built upon these two operating systems

Environmental Modeling Using Open Source Tools, Table 2 Web-based FOSS geospatial software stack based on FOSS tools

Generic stack	Geospatial software stack	Grouping
Interactive web mapping application	Custom Java applications uDig, Firefox	Web-based environmental DSS
	Internet	
Application development tools	MapServer, MapGuide Open Source, MapBender	User interface and visualization tools
	PyWPS	Web processing services
	Apache http server, Content management systems	Document serving
Scripting languages	Perl, Python, R	Data processing, data management, data serving, data integration
Libraries, command line utilities	GeoTools, PostGIS, Proj4, GDAL/OGR, JTS/GEOS, GMT	
Programming languages	C, C++, Java, C#, FORTRAN	System software
System libraries	Network and data exchange libraries	
OS kernel	Linux, Solaris, MS-Windows	

and easy as, for example, spreadsheets, as development platforms. Each platform presents challenges to the GIS developer. For example, the desktop platform allows for more intensive use of local disk space, memory, and processing power than does a web-based platform. Hence, on one hand, the desktop is better for large, computationally intensive applications that use large data sets and require complex user interaction. While on the other hand, web-based applications generally have the advantage of being more rapidly deliverable to end users, more easily updated, and more centrally controllable. An interesting compromise occurs when a desktop application is developed to be "web-aware" or "web-enabled". In this case, the user gains the benefit of local data processing and storage while using the internet to download software updates and share data or models with a wider user community.

Geospatial Software Stacks for Environmental Modeling and Management

A high level of interoperability makes FOSS very interlinked or "layered" such that most tools tend to depend on other lower-level tools. Layering results in "software stacks," which can be very deep, the layers at the bottom being, for example, the Linux kernel or the GNU C library, libc. FOSS coexists with and adjusts to proprietary software platforms easily. Examples include MinGW, which implements the GNU platform partially in Microsoft Windows, and Java, which is at least partly a proprietary platform of Sun, but also a popular FOSS language. Thus, a software stack for geospatial work may comprise FOSS, but also some proprietary products. A typical desktop FOSS4G stack and a typical web-based FOSS4G stack are shown in Tables 1 and 2; every tool listed is FOSS except for the two proprietary operating systems, Mac OSX and MS-Windows. Any software stack used to solve a real geospatial EMM problem will necessarily vary from these examples as it adjusts to unique requirements. Somewhat similar diagrams than these tables have been presented by Ticheler [7]. Key parts of the stacks are described below.

System Software System software is the foundation upon which a complete software stack is built. It provides basic interoperability and some common tools. Linux and GNU provide a common UNIX-like operating system (OS) platform for geospatial software tools. The Microsoft Windows OS can also be used, and with the .NET framework it provides a platform for developing easily extensible systems. System software layers in the two stacks presented here contain many common items. This synergy between the software stacks shows great promise in intertwining capabilities. For example, it can be envisioned that future desktop applications will rely more on web services, while web-based applications will contain more of the functionality traditionally relegated to the desktop platform.

Geospatial Data Processing Libraries In the data processing layer, data or essential information about it is retrieved into the system memory for processing. Data access and usage details are often tool specific. A benefit of FOSS4G is unobstructed access to these details. Data processing is divided here into data management, data integration, and geoanalytical processing. The foundation of the domain-specific interoperability of the geospatial tools is in this layer. Solving problems in EMM requires complex workflows that involve interoperation of several tools in order to be effective.

Data Management Data management is a critical function of GIS for EMM. Both the number of required geospatial datasets and their size are often voluminous. Geospatial datasets are stored within specialized file formats or databases using either a vector- or raster-based model. Geospatial data sets are often provided for use from servers either through networked file systems or via the internet. Attributes and metadata are stored and managed along with the data or separately in spatial databases. A relational database management system (RDBMS) is a platform for developing applications, functionality, and services. On the FOSS platform, the two main RDBMSs that support the Structured Query Language are MySQL and PostgreSQL. A general perception is that MySQL is less featured (implements less of the SQL standard) than PostgreSQL, but faster. Thus MySQL is probably more popular in web-based solutions and PostgreSQL in desktop solutions. Geospatial data are not easy to store in a standard RDBMS, thus spatial extensions have been developed. PostGIS is an add-on to PostgreSQL, which provides the standard spatial data types and spatial queries. Spatial data handling capabilities have also been added to MySQL recently.

Data Integration Environmental modeling often requires integration of various geospatial data from multiple sources. An important component of the geospatial software stack is a tool to transform datasets from one geospatial coordinate system to another. The most common FOSS4G tool for this is PROJ.4. For example GRASS [8], QGIS, and MapWindow GIS use PROJ.4 for their map projection capabilities. PROJ.4 contains a system for describing common spatial reference systems and for converting data between projections. The GDAL/OGR library and associated utility programs provide widely used basic functionality on the FOSS platform. GDAL provides a generalized Application Programming Interface (API) for raster data and a way to implement drivers for raster data formats and OGR does the same for vector data. The GDAL/OGR library is written in C++ and it has been ported to several common operating systems. The OGR library can optionally be compiled to directly include the GEOS library, which adds standard two-dimensional spatial predicates and functions. A set of common general-purpose tools have been implemented on top of the GDAL/OGR API. Bindings for scripting languages like Python, Perl, and Ruby to the GDAL/OGR library also exist.

Geoanalytical Tools A number of FOSS libraries support numerical calculation, geostatistics, and computational geometry operations. The basic analytical method family for raster datasets is map algebra (a term coined by Tomlin [9]), which extends the standard algebra of scalar values to raster data. Map algebra extended with hydrological functions is directly usable in environmental modeling [10]. Tools for spatial modeling are implemented within desktop GIS as fully integrated modules, plug-ins, or as stand-alone applications.

Data Storage and Serving The data-serving layer exists mainly in the web platform as tools that receive data from the data-processing layer and serve it to the user interface layer. Data storage and serving can be managed by private or public entities using a variety of proprietary and FOSS tools. The data-serving layer is important since it enables a wholly new type of software, such as collaborative applications that are valuable in EMM. These tools can be used to serve maps (as images) and thus create interactive websites, but they can also be used to serve data directly to custom applications according to web mapping service (WMS) and web feature service (WFS) standards.

Scripting Languages Scripting languages are general-purpose programming languages with an associated application, which immediately executes the code. Scripting languages can be extended with modules, and collections exist, which contain modules for data access, graphics, mathematical computations, networking tools, to name a few areas. Scripting languages are very useful in "gluing" tools together to create EMM solutions. Rapid development and interactive development tools make scripting languages attractive for "user-developers" [11]. Bindings, which enable geospatial libraries to be used from scripting languages, are increasingly available.

User Interfaces The user interface is often, but not necessarily, significantly different between the desktop and web platforms. Modern desktop GIS tools focus on allowing the user to create maps for a particular purpose from specific geospatial data. The web platform is currently dominated by the web browser as the ubiquitous client application. Some new tools have begun to blur the distinction between a desktop and web end-user application. The enabling technology is the standards [Open Geospatial Consortium (OGC)]-based serving of geospatial data. Ease of use through a GUI can be important for EMM software end users. However the best GUIs also provide modelers with powerful tools for developing and editing data sets, implementing models, making visualizations, etc.

End-User Applications At the top of the FOSS software stack are end-user applications: the tools that are used by managers, modelers, stakeholders and others to better understand the environment. These can be as simple as a customized map viewer on a website showing the location of a particular environmental problem, or as complex as a fully integrated dynamic simulation model of a watershed integrated with a desktop GIS. In all cases, it is at this top layer where end users interface with the software. Ultimately, the needs and requirements of the EMM community have to be met at this layer. As Maguire [12] concludes, while proprietary software dominates in GIS, FOSS has an upper hand in modeling software. For example, notable environmental simulation models SWAT, HSPF, WASP, and QUAL2E are FOSS. FOSS4G interfaces for many of these models have been developed or are under development.

Key Applications

Several efforts to merge environmental modeling with FOSS4G tools have been initiated at the United States Environmental Protection Agency (US EPA) Office of Science and Technology (OST), US EPA Office of Research and Development (ORD), and the United States Department of Agriculture (USDA). In each case an existing environmental model or modeling system is being adapted to explicitly use geospatial data in a FOSS4G environment. Specifically, a project to migrate the BASINS watershed modeling system to MapWindow GIS was initiated by OST. Similarly ORD is currently investing in the adaptation of MapWindow GIS, GDAL, JTS, and other FOSS4G tools to support a wide variety of EPA environmental modeling tools, beginning with the FRAMES/3MRA system. USDA through its Texas A&M university collaborators are following suit with the development of a new GIS interface to its SWAT watershed model, again based on FOSS4G tools. With respect to web-based applications, the USGS StreamStats tool (http://water.usgs.gov/osw/streamstats/) is an example system that uses a combination of proprietary and FOSS elements to make a publicly accessible and highly functional EMM tool for watersheds.

Environmental management can be divided into a learning phase and a decision-making phase. Descriptive and simulation models are used for learning, but sometimes simply browsing existing information suffices. Maps are particularly good tools for providing information needed in both of these phases. A set of alternative actions, along with impact assessment have to be devised for the decision making phase. The alternatives should be presented such that decision makers better understand the potential consequences of specific actions. Simulation models are routinely used for impact assessment. Optimization models can be used to suggest decisions once evaluation criteria and methods for computing them are selected. All of these model types and uses can benefit from integration with GIS, and particularly FOSS4G GIS, because of its open and transparent nature.

Two additional key applications of FOSS4G for EMM include environmental simulation models and environmental management applications. In the case of simulation models, pre- and postprocessing of geospatial data is often required and a GIS can provide both the data processing and GUI capabilities. Harvey and Han [13] have presented an excellent analysis of the relevance of FOSS to hydraulic and hydrological simulation models where a FOSS4G GIS is used in this manner. In the case of environmental management, FOSS4G tools can be used to aid in monitoring the state of the environment, planning of actions for improving that state, and increasing people's awareness of it. Transparency and interoperability are key FOSS4G features that help to address these needs.

Environmental management projects may be large and international and hence require ease of data sharing, interoperability, inexpensive implementation, and transparency of all model and data-management software. For these and other reasons, many international organizations and initiatives such as UNESCO (United Nations Educational, Scientific and Cultural Organization) and FAO (Food and Agriculture Organization of the United Nations) [7] are currently investigating FOSS4G for their needs in developing GIS-based EMM systems.

Future Directions

As computing becomes more ubiquitous, the significance of one single tool becomes less important and focus shifts more towards software stacks and platforms. The dividing line between FOSS and proprietary software is fuzzy, partly because it is in the interest of developers of proprietary software to make it fuzzy and partly because end users are becoming more reluctant to buy software. People expect web browsers and other common tools to be free of charge. Also, depending on license of the particular FOSS tool, proprietary tools may include FOSS. Advances in standards aiming for interoperability and the mixing of platforms make it possible to co-use free and proprietary tools. FOSS4G targets in many cases advanced users and developers. This is consistent with the needs of EMM since the latter field is also dominated by engineers who often take the role of both user and developer. The barriers to adoption of FOSS4G solutions for the EMM application areas are not many. One barrier may be that FOSS4G tools can sometimes be less well "packaged" than proprietary products. However companies do exist that specialize on delivering complete FOSS4G solutions. It is incumbent upon developers of FOSS4G tools to improve the ease of use, installation, and integration of such tools so that they can be more readily adopted by the environmental modeling community. Potential improvements might include:

- Providing compiled binaries for multiple platforms and operating systems
- Developing demonstration applications that show how one integrates FOSS4G tools with existing environmental models
- Generating simplified installation packages that can be readily adapted and integrated with the installation package of a particular model
- Enhancing existing user communities and developing new discussion forums targeted specifically at FOSS4G users in the environmental modeling community
- Clarifying the meaning and interpretation of various FOSS license agreements
- Seeking out opportunities to adapt FOSS4G tools to the specific needs of the EMM community

As these and other advancements are made, it is expected that FOSS4G-based implementations of many EMM models and tools will become more readily available, directly benefiting engineers, managers, modelers, developers, and EMM stakeholders alike.

Recommended Reading

1. Ramsey, P.: The State of Open Source GIS. http://www.refractions.net/white_papers/oss_briefing/2006-06-OSS-Briefing.pdf. Accessed 24 Aug 2006
2. Kovar, K., Nachtnebel, H.P.: Application of geographic information systems in hydrology and water resources management. In: Proceedings of the HydroGIS'96 Conference, Vienna, 16–19 Apr 1996
3. Goodchild, M.F., Steyaert, L.T., Parks, B.O. (eds.): GIS and Environmental Modeling: Progress and Research Issues. GIS World Books (1992)
4. Mitasova, H., Mitas, L., Brown, W.M., Gerdes, D.P., Kosinovsky, I., Baker, T.: Modeling Spatially and Temporally Distributed Phenomena: New Methods and Tools for GRASS GIS. Int. J. Geogr. Inf. Sci. **9**(4), 433–446 (1995)
5. Frysinger, S., Copperman, D.A., Levantino, J.P.:. Environmental decision support systems (EDSS): an open architecture integrating modeling and GIS. In: Goodchild, M.F., Steyaert, L.T., Park, B.O., Johnston, C., Maidment, D., Crane, M., Glendinning, S. (eds.) GIS and Environmental Modeling: Progress and reasearch Issues, pp. 357–362. GIS World Books, Fort Collins, CO
6. Anonymous: The free software definition. http://www.fsf.org/licensing/essays/free-sw.html (2006). Accessed 26 Apr 2006
7. Ticheler, J.: SDI Architecture diagram. http://193.43.36.138/relatedstuff/index_html/document_view Accessed 10 Mar 2006
8. Neteler, M., Mitasova, H.: Open Source GIS: A GRASS GIS Approach, 3rd edn. Springer, NY (2007)
9. Tomlin, C.D.: Geographic Information Systems and Cartographic Modeling. Prentice-Hall, Englewood Cliffs, NJ (1990)
10. DeMers, M.N.: GIS Modeling in Raster. Wiley, New York (2002)
11. Rosson, M.B.: The end of users. In: Proceedings of the OOPSLA (2005) Conference (keynote presentation), San Diego, California, October 16–20, 2005
12. Maguire, D.J.: Towards a GIS platform for spatial analysis and modeling. In: Maguire, D.J., Batty, M., Goodchild, M.F. (eds.) GIS, Spatial Analysis, and Modeling, pp. 19–39. ESRI Press, Redlands, California (2005)
13. Harvey, H., Han, D.: The relevance of open source to hydroinformatics. J. Hydroinformatics **4**, 219–234 (2002)

Environmental Planning and Simulation Tools

DALE A. BRUNS
College of Science and Engineering, Wilkes University, Wilkes-Barre, USA

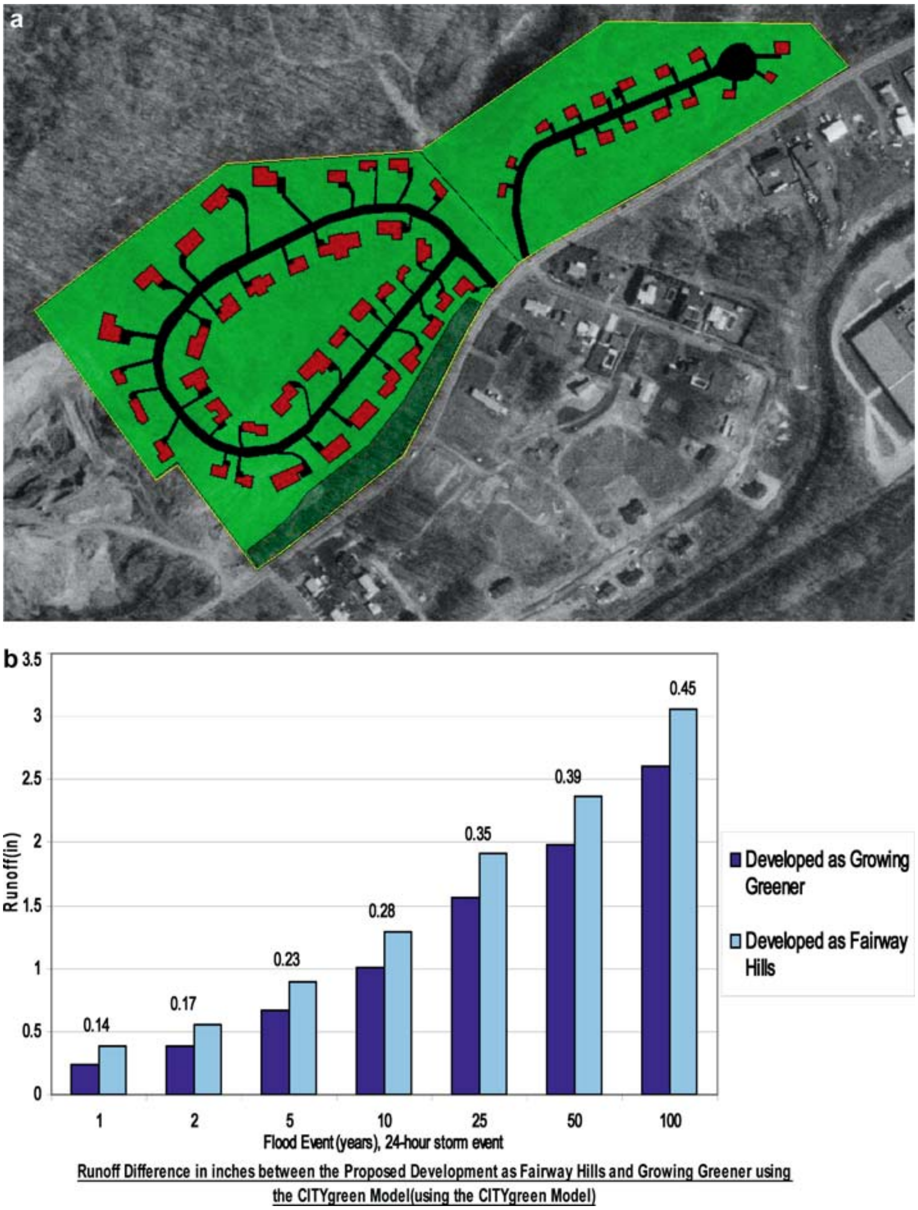
Synonyms

Land use planning and environmental modeling; Ecological planning and modeling; Environmental planning support systems

Definition

Environmental planning is an interdisciplinary, problem-solving approach to alleviating impacts from human activities within ecosystems usually in regard to preventing damage to ecological resources or ameliorating impacts to human health, well-being, or quality of life. Geospatial data, software, and models or simulations are used as tools

E



Environmental Planning and Simulation Tools, Figure 1
a Two scenarios analyzed with CITYgreen [1]: on the left is a conventional housing development (Fairway Hills) with most trees removed and standard-size yards, driveways, access roads, and building footprints; on the right is a "clustered" development (Growing Greener design) with original trees maintained as a greenway (open space) maintained for community use with a minimum of impervious surfaces like building roofs, roads, and driveways
b Graphic output from CITYgreen modeling of stormwater runoff (in inches, already corrected for area) as a function of various 24-hour storm events (1-year to 100-year probability events) for conventional (Fairway Hills) development vs. "clustered" (Growing Greener) environmental design [1]. In a one year event, 20,000 gallons of water are "conserved" with the environmental design while over 180,000 gallons of potential stormwater are "removed" from a 100-year storm event with the Growing Greener approach [1]

to facilitate the planning process on a team-basis among communities and their leaders, the business sector, and various specialists in technical areas ranging from engineering, geology, geography, hydrology, and ecology to natural resource disciplines such as forestry and fisheries. The goal of environmental planning is to balance economic, social, and ecological factors in a sustainable manner to the long-term benefit of livable communities, economic development, and ecological renewal of natural resources. The latter includes but is not limited to clean air and water, and naturally functioning ecosystems such as forests, streams, lakes, and coastal ecosystems. Geographical information systems (GIS) and geospatial data (e. g., remote sensing imagery and ortho-photography) are the predominant tools that allow planners to develop visual models (or maps) of the human environment (e. g., roads, buildings, parking lots, property boundaries) and components of associated natural ecosystems. Applications of GIS can be used to map spatial and functional relationships between human and natural environments and, based on model simulations, predict influences between them, and provide alternative scenarios for best practical solutions that promote sustainability. Figure 1 provides a GIS map that illustrates two alternative designs for residential developments relative to natural stands of trees [1]. A geospatial model that incorporates factors such as slope, land cover (man-made vs. natural), and soil permeability allows for the simulation of storm-water runoff volumes. This simulated GIS model of the environment provides the residential community, developer, and planner with contrasting planning scenarios, and includes both economic costs and benefits, to be considered for implementation in the context of long-term sustainability.

Historical Background

Environmental planning has evolved from different historical perspectives. Federal legislation such the Clean Water Act (CWA) in 1972, the Federal Land Management Act of 1976, and the Endangered Species Act (1973), have elicited various planning approaches that cut across traditional political boundaries. For example, a federal plan for managing the spotted owl and its habitats has encompassed public lands within several watersheds and numerous political units in the states of both Washington and Oregon [2]. Likewise, the CWA requires watershed management plans that range from regional coverage to localized tributary plans developed by community groups. This branch of environmental planning, focused on natural resource management, may also embody national, continental and global plans by international non-profit organizations such as the Wildlife Conservation Society [3], the Nature Conservancy [4], and the World Wildlife Fund [5]. Such plans provide a basis for managing wild-lands for conservation and recreation; local land trusts may also take a similar approach [6].

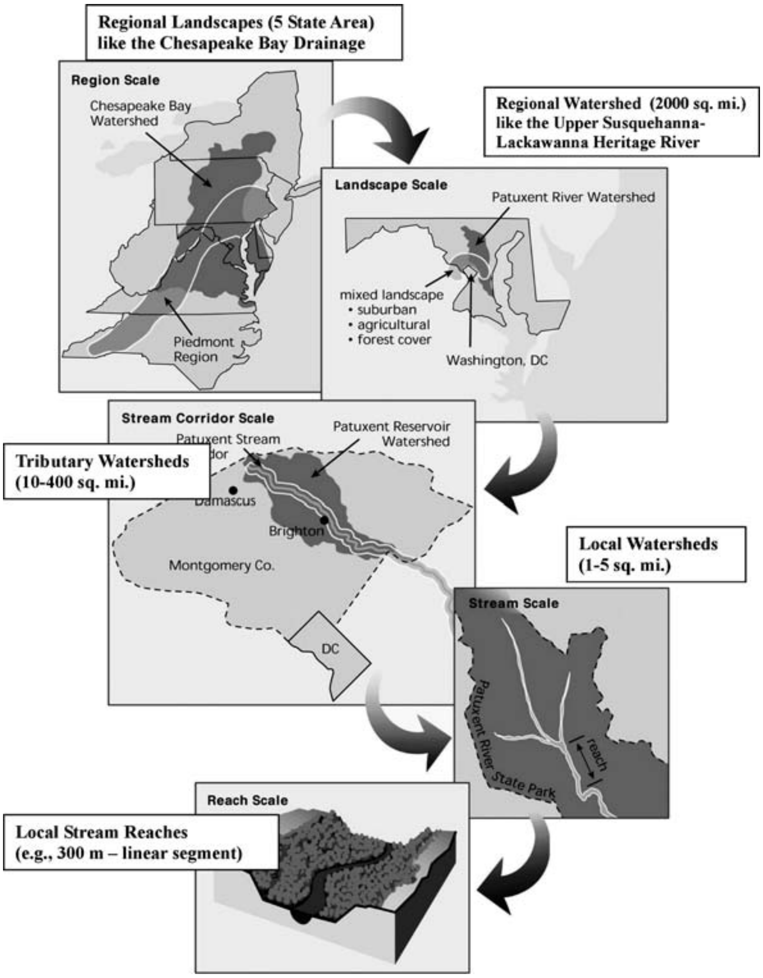
Environmental land use planning emerged from a vision of development activity based on a balance among ecological, social, and economic factors. For example, environmental designers such as Olmsted, Howard, McHarg, and Corbett incorporated the concept of "greenspace" or undeveloped natural open space in their planning projects in the United States prior to 1980 [7]. McHarg [8] in particular foreshadowed the emergence of computerized geographic information systems with his use of hard copy "map overlays" for visualizing development alternatives that allowed for evaluation of environmental factors such as slope, soils, vegetative cover, and hydrology – along with cultural features such as utilities and transportation.

Environmental planning, at various spatial scales, and including both natural resource and land use planning, converges on several historically unifying components: 1) use of digital data layers, derived from aerial photography and satellite remote sensing imagery, 2) a reliance on GIS for mapping, analysis, and modeling, and 3) an overarching goal of sustainability. Applications of geospatial data and GIS developed during the 1980s, and became widely affordable and typically used through the middle and late 1990s by local government, state and federal resource management agencies, and non-profit conservation groups. The concept of sustainable development or "sustainability" emerged in the 1980s and was defined in a report by the World Commission on the Environment and Development [9] as "an approach toward meeting the needs and aspirations of present and future generations without compromising the ability of future generations to meet theirs." Some environmental planners suggest that the roots of this concept, related to a "land ethic," can be traced to the writings of Leopold and Thoreau [7].

Scientific Fundamentals

The planning process for environmental issues vary in regard to spatial (and temporal) scales and the relative balance between a primary focus on development vs. a resource management emphasis in the context of policy, statutes, regulations, or public advocacy from non-profit organizations and community groups. Nevertheless, the environmental planning process [7] typically includes the following elements: 1) public scoping and identification of issues, 2) data inventory of existing conditions, 3) definition of conflicting issues (e. g., development vs. a balance with concurrent ecological factors), 4) analysis of problems, 5) modeling and assessment of alternative scenarios or solutions (often with cost-benefit analysis), 6) implementation with monitoring and assessment, and 7) public outreach, education, and participation. Although these steps may serve as an approximate sequential timeline, several of these elements often proceed concurrently and are iterative in nature [7]. In addition, the "unit of study" for environmental planning now more often reflects natural ecological boundaries such as a watershed as opposed to conventional political delineations historically associated with economic development in isolation from an environmental context [7,10,11]. These environmental planning elements, within a watershed approach, have been illustrated for an American Heritage River (AHR) watershed and community program [12,13].

Public participation is an integrating component throughout the planning process, from scoping and problem identification to implementation of the plan. For the AHR watershed [12], a series of outreach meetings among local government representatives, federal and state agency officials, industry participants, university scientists, local environmental groups, and the general public identified and prioritized the extent and importance of regional environmental problems with an estimated $2.5 billion cleanup [13]. These issues ranged from abandoned mining lands to problems of water quality and flooding, along with local data acquisition needs to support government applica-



Environmental Planning and Simulation Tools, Figure 2 Ecosystems at multiple scales and used as the basis for regional to local GIS watershed analysis and planning for the Upper Susquehanna–Lackawanna American Heritage River [11,12]. Modified from the Interagency Stream Restoration Manual [15]

tions, including conservation, environmental restoration, economic planning, and emergency response [13].

Team leaders for environmental planning often rely on existing geospatial databases, that are typically incomplete or of limited spatial resolution, to identify and assess current problems, develop alternative scenarios and solutions, and establish baseline conditions to monitor benefits of implementing the program [12]. In the AHR illustration, an interdisciplinary team (e. g., geochemist, river ecologist, engineer, and community leaders) from various institutions and agencies was assembled to propose, discuss, evaluate, and recommend a GIS watershed approach to regional planning, monitoring, and assessment; this included an ecosystem conceptual framework [11] that allowed for spatial scaling from local, site-specific and stream-reach applications (e. g., well-suited to local watershed groups) to broader watershed and landscape spatial scales – the latter represented by federal agency geospatial data and analysis [14,15], see Fig. 2. In this context, tributary sub-watersheds were delineated within a GIS analysis to rank and prioritize problem sites relative to identified environmental problems and a rationale for reclamation [11,12,13].

In general, an environmental plan can be employed as a long-term strategy document that provides a "road map" for state and federal agencies, local communities and governments, and non-profit organizations (e. g., regional universities and land trusts) in solving or preventing problems of the environment. As an example, the GIS watershed analysis for the AHR facilitated implementation of the plan with follow up projects intended to benefit the more than 190 local entities within the 2000 square mile planning area [13]. Implementation of the GIS Environmental Master Plan [12] included the following emergent activities:

- Local-to-federal resource leveraging (486%) to obtain local scale aerial data valued at over $2M [13];
- Demonstration projects with CITYgreen – for suburban "buildout" or "spawl" in a planned development [1]; in a tributary watershed with severe flooding prob-

lems [16]; and on the benefits of reforestation on mining lands relative to carbon sequestration [17];
- Real-time water quality monitoring of the river [17];
- Web-based GIS for public dissemination of environmental information [13,17];
- County sharing and distribution of aerial data via Web based GIS [13];
- Technical support and coordination among community partners for environmental education [12,13,16];
- 3-D visualization (CommunityVIZ) and simulation of a local borough with zoning for ecological resources and eco-tourism [13,17]; and
- Participation in a university GIS research program (U.S. Department of Agriculture funded) as part of a national consortium focused on rural, place-based planning support systems [18].

Key Applications

Natural Resource Planning

GIS mapping and analysis encompass a wide range of ecological applications including endangered species, clean water, aquatic pollution, forestry and wildfires, air pollution, and disaster planning and recovery from floods [19]. Environmental (and cultural) data for such uses also reflect a broad range of spatial scales from local to regional to global in scope [15], see Fig. 2. The following are representative examples of this approach.

Mapping the Global Human Footprint

The Wildlife Conservation Society [3] has employed datasets on human population densities, land transformation, access corridors (e.g., roads, railways, coastlines), electrical power infrastructure, and terrestrial biome extents to "simulate" the collective potential for anthropogenic affects on wildlife and "wild" habitat worldwide. A relative scoring system was used as a "human influence index" that was mapped as a gradient against a visualization of biomes on a global basis. Similar applications and analysis on a global basis have been conducted for related conservation planning efforts by other international organizations [4,5].

Regional Landscape Assessment

The U.S. Environmental Protection Agency's Mid-Atlantic Regional Landscape Atlas [14] provides an assessment of land use relative to agricultural and urban impacts such as nutrient loading and loss of natural habitats. This study employed GIS to "simulate" 33 indicators to reflect environmental conditions in a five state area, and has served as a baseline for local watershed planning [12].

Regional Wildlife Management Planning

Environmental statutes and regulations for management of public lands require land use planning such as "Forest Plans" under the purview of the U.S. Department of Agriculture Forest Service. An example of wildlife population and habitat management as part of a forest plan [2] is that for the spotted owl in the northwestern U.S.

Gap Analysis

"Gap analysis" is a geographical approach to protection of biological diversity [20] and has been shown as an effective tool for strategic conservation planning in the U.S. [21]. Gap analysis "superimposes species distributions with boundaries of ecosystems and protected areas to identify gaps in the protection of species" [21]; analyses have been employed in California [22], Utah [23], Florida [24] and Costa Rica [25].

BASINS

Better Assessment Science Integrating Point and Nonpoint Sources (BASINS) is a "multipurpose environmental analysis system for use by natural resource agencies to perform watershed- and water-quality-based studies and to provide a framework for examining management alternatives" [26,27]. Multimedia data have been compiled for GIS data mining and other assessment tools.

SWAT

The Soil Water Assessment Tool (SWAT), originally developed [28] for management and assessment of agricultural and rangelands, is a GIS model that incorporates data on climate, rainfall, soils, slopes, and land cover; SWAT addresses changing land use with runoff and sediment yield as watershed model outputs [29].

CITYgreen

CITYgreen provides "analysis on the environmental function and economic value of trees and forests," especially in urban areas [30]. This planning tool uses either local tree data or satellite imagery for land cover, along with data on soils, slopes, and rainfall. Model output includes runoff, carbon sequestration, and air pollution removal. CITYgreen has predicted the outcome of various development scenarios based on land use change [1]. In addition, CITYgreen has been used to evaluate the potential for carbon sequestration if barren mining lands were to be reforested [17]. It has also been profiled as a watershed planning tool [16] and evaluated as a planning support system [31].

ModelBuilder

ModelBuilder is a general graphic design tool for "diagramming solutions to spatial analysis problems" [32] and can serve as a basis for natural resource management. A brief introduction to ModelBuilder is available in an educational workbook [32] and in modeling of water resources [33].

Planning Support Systems

Planning support systems (PSS) comprise "that constellation of digital techniques (such as GIS) which were emerging to support the planning process" [34]. As such, PSS are not highly structured computer models, but rather "loosely coupled assemblages of computer-based techniques" – which form a "toolbox" for decision makers in framing questions for problem solving in community planning [34]. Several definitions of PSS have been reviewed in *Environmental and Planning B: Planning and Design* [35]. PSS have been considered from the perspective of various GIS models and applications at conferences [36], and in books [37,38], reports [39] and reviews [31,35]. In general, research on PSS is fairly extensive, while use by professional practitioners is more limited [35]. Several PSS are reviewed below.

CommunityVIZ

CommunityVIZ is a suite of GIS-based modules including Scenario Constructor, SiteBuilder3D, and Policy Simulator [40]. The software is integrated so that changes to data in one module are reflected automatically in the others, allowing edits and providing results in 2-D and 3-D formats for evaluating alternative policies over time [40]. For example, Policy Simulator is an "agent based" model where "individuals" are units of analysis and output is "bottom-up" and stochastic to predict probable land use, demographics, and economic changes [40,41].
CommunityViz has been discussed in a general volume on PSS [38] and has been evaluated by a USDA-funded university research consortium focused on rural GIS applications [18,31], including indicators of increased water consumption due to a proposed ski resort [42,43], visualized 3-D models of land use impacts from mining [17], and planning for watersheds [13].

What if?

What if?, designed for collaborative planning and decision making, employs available GIS datasets and encompasses the three major components of land-use planning: suitability, growth, and allocation [44]. Its GIS procedures allow for "suitability analysis," projecting future land use demands, and allocating such demands to suitable locations. Overviews of this PSS are provided in journal [45] and book [44] publications and it has been used to evaluate farmland preservation policies [46] and a sustainable urban-growth land use strategy for an Australian coastal city [47]. What if? has been evaluated in pilot projects by a USDA-funded university research consortium [18,31] and is one of 22 land use change models evaluated by the U.S. EPA [39,48].

STEPP

This research tool incorporates environmental aspects in planning and is intended to facilitate local progress in creating a sustainable environment [49]. STEPP derives "zones of influence" based on where environmental impacts occur. Three categories of a "receptive environment" are delineated: 1) areas with human inhabitants, 2) areas with unique ecological values (e. g., nature areas), and 3) areas without unique ecological values but subdivided by soils and water. This PSS, developed in the Netherlands, is intended to be a user-friendly environmental analysis tool for local authorities [49].

Planning Analyst

This PSS is a toolkit with "software modules that support geographic data exploration and analyses, land use allocation, and impact assessment," and includes 3-D mapping and generation of future growth scenarios [50]. A primary goal was to provide an approach, with minimal training, that would allow citizens and groups to create and evaluate their own planning alternatives. One component, Place-It, "allows users to interactively allocate land uses to parcels of land," with statistics on land conversion, population changes, and fiscal and environmental impacts [50].

Community Planning Through Web-based Technologies

Internet capabilities are expanding the potential for use of PSS by local communities in several ways. First, the Internet allows planners to obtain updated information about different PSS [35]. For example, a "Planning Portal" has been constructed on-line that provides information on resources for community planning, plus geospatial tools for a local community [50]. Second, internet map software, such as ESRI's ArcIMS [11,50], allows local government to maintain local ortho-imagery and parcels for public access to maps relevant to community planning [13,17]. And third, GIS tools now allow "planners, decision-makers, and members of the public" to interrogate databases to perform "spatial and non-spatial queries through accessing a dedicated on-line planning support system" [51].

Future Directions

Expanded web-based capabilities of "user-friendly" geospatial databases and GIS planning tools are expected in the future [11,13,50,51]. It is anticipated that more academic researchers will apply PSS to "real projects" managed by practicing planning professionals [35,47]. Natural resource managers and environmental planners are predicted to employ more 3-D GIS analyses, such as the visualization tools in CommunityVIZ [13,17,41,42,43,50], since these are helpful in public outreach and understanding of alternative planning scenarios. Federal, state, and local community planning will need to address issues of emergency preparedness and response regarding terrorists' threats to homeland security [11,52,53], including ecological impacts to water, air, and the food chain (e. g., nuclear, chemical, biological agents – dispersed through environmental pathways to human and ecological endpoints). And finally, global warming and climate change may emerge as a major environmental planning concern (local to global), including pressures on homeland security [54].

Cross References

- ► ArcGIS: General Purpose GIS Software System
- ► Change Detection
- ► Decision-Making Effectiveness with GIS
- ► Distributed Hydrologic Modeling
- ► Evolution of Earth Observation
- ► Homeland Security and Spatial Data Mining
- ► Internet GIS
- ► Multicriteria Decision Making, Spatial
- ► Rural Hydrologic Decision Support
- ► Scale, Effects
- ► Uncertain Environmental Variables in GIS
- ► Visualizing Constraint Data

Recommended Reading

1. Carl, J.R., DeAngelo, M.M., Ruskey, D.C.: Sustainable development plan and runoff study for a proposed development in the Hick's Creek watershed. In: Montz, B.E., Tobin, G.A. (eds.) Papers and Proceedings of the Applied Geography Conferences, vol. 25, pp. 9–17. Binghamton, NY (2002)
2. Link, J.: Technical Coordinator: Northwest Forest Plan – the first 10 years (1994–2003): status and trends of northern spotted owl populations and habitat. General Technical Report PNW-GTR-648. Portland, OR. U. S. Department of Agricultures, Forest Service, Pacific Northwest Research Station. 176 p (2005)
3. Sanderson, E.W., Jaiteh, M., Levy, M.A., Redford, K.H., Wannebo, A.V., Woolmer, G.: The human footprint and the last of the wild. Bioscience **52**:891–904 (2002)
4. Sawhill, J.C.: Creating biodiversity partnerships: the Nature Conservancy's perspective. Environ. Manag. **20**:789–792 (1996)
5. World Wildlife Fund: The Living Planet Report 2006. WWF, Switzerland (2006)
6. Land Trust Alliance: About LTA: Local Land Trust Success Stories. Retrieved December 23 (2006) from http://www.lta.org/
7. Randolph, J.: Environmental Land Use Planning and Management. Island Press, Washington, DC (2004)
8. McHarg, I.: Design with Nature. American Museum of Natural History (1969)
9. World Commission on Environment and Development: Our Common Future. Oxford University Press (1987)
10. DeBarry, P.A.: Watersheds: processes, assessment, and management. John Wiley and Sons, Inc., Hobokon NJ (2004)
11. Bruns, D.A., Wiersma, G.B.: Design of environmental monitoring systems: a geospatial perspective. In: Wiersma, G.B. (ed.) Environmental Monitoring, pp. 1–35. Lewis Publishers (2004)
12. Bruns, D.A., Sweet, T., Toothill, B.: Upper Susquehanna-Lackawanna River Watershed, Section 206, Ecosystem Restoration Report. Phase I GIS Environmental Master Plan. Final Report to U.S. Army Corps of Engineers, Baltimore District, MD (2001)
13. Bruns, D.A., Sweet, T.: PPGIS in Rural America: Integration of Geospatial Technologies and Resource Leveraging in Data Acquisition for an American Heritage River community. In: Proceedings of the Annual PPGIS Conference of the Urban and Regional Information Systems Association (URISA), pp. 43–70. University of Wisconsin, Madison, WI (2004)
14. Jones, K.B., Riitters, K.H., Wickham, J.D., Tankersley Jr., R.D., O'Neill, R.V., Chaloud, D.J., Smith, E.R., Neale, A.C.: An Ecological Assessment of the United States Mid-Atlantic Region: A Landscape Atlas. U.S. EPA, Office of Research and Development. Washington, DC. EPA/600/R-97/130 (1997)
15. Interagency Team: Stream Restoration Guidance Manual. Stream Corridor Restoration: Principles, Processes, and Practices. By the Federal Interagency Stream Restoration Working Group (FISRWG) (15 Federal agencies of the US government). GPO Item No. 0120-A; SuDocs (1998) ISBN-0-934213-59-3
16. Bruns, D.A.: Land use planning with CITYgreen: GIS software evaluates development impacts on "green" infrastructure. Land Bulletin of the National Consortium for Rural Geospatial Innovations. University of Wisconson, Madison, WI. www.ruralgis.org In Press (2007)
17. Bruns, D.A., Sweet, T.: Geospatial tools to support watershed environmental monitoring and reclamation: assessing mining impacts on the Upper Susquehanna-Lackawanna American Heritage River. U.S. Department of Interior, Office of Surface Mining. Conference Proceedings, Advanced Integration of Geospatial Technologies in Mining and Reclamation, December 7–9, Atlanta, GA. http://www.tips.osmre.gov/GeoSpatial/post-conference/Papers/Bruns_Sweet1_PA.pdf
18. National Consortium for Rural Geospatial Innovations: Mission, Goals, and Publications at www.ruralgis.org
19. Lang, L.: Managing Natural Resources with GIS. ESRI Press (1998)
20. Scott, J.M., Csuti, B., Jacobi, J.D., Estes, J.E.: Species richness: a geographic approach to protecting future biological diversity. Bioscience **37**:782–788 (1987)
21. Savitsky, B.G.: Overview of Gap Analysis. In: Savitsky, B.G., Lacher Jr., T.E. (eds.) GIS Methodologies for Developing Conservation Strategies., pp. 151–157. Columbia University Press (1998)

E

22. Davis, F. W.: Gap analysis for the southwestern California region. Technical Report 94–4. National Center for Geographic Information and Analysis, Santa Barbara, California (1994)
23. Edwards Jr., T.C., Homer, C.G., Bassett, S.C., Falconer, A., Ramsey, R.D., Wight, D.W.: Utah Gap analysis: an environmental information system. Final Project Report 95–1. Logan: Utah Cooperative Fish and Wildlife Research Unit (1995)
24. Cox, J., Kautz, R., MacLaughlin, M., Gilbert, T.: Closing the gaps in Florida's wildlife habitat conservations systems. Office of Environmental Services, Tallahassee, Florida (1994)
25. Savitsky, B.G., Lacher, T.E., Jr. (eds.): GIS Methodologies for Developing Conservation Strategies. Columbia University Press (1998)
26. Environmental Protection Agency (EPA), Better Assessment Science Integrating Point and Nonpoint Sources (BASINS): User's Manual, Version 3.0, U.S. EPA Office of Water (4305), Washington, D.C., EPA-823-H-01-001, http://www.Epa.gov/ost/basins (2001)
27. Partington, E., Wells, D., Manguerra, H.: What's new with EPA/Office of Water's BASINS Water Quality Modeling System. ESRI User's Conference. San Diego, CA http://gis.esri.com/library/userconf/proc01/professional/papers/pap909/p909.htm (2001)
28. Srinivasan, R., Arnold, J.G.: Integration of a Basin-Scale Water Quality Model with GIS. Water Resour. Bull. **30**:453–462 (1994)
29. Hernandez, M., Miller, S., Goodrich, D.C., Goff, B.F., Kepner, W.G., Edmonds, C.M., Jones, K.B.: Modeling runoff response to land cover and rainfall spatial variability in semi-arid watersheds, Environ. Monit. and Assess. **64**:285–298 (2000)
30. American Forests: CITYgreen: Calculating the Value of Nature, Version 5.0 User's Manual, American Forests, Washington, D.C., See also http://www.americanforests.org/productsandpubs/citygreen/ (2002)
31. Niemann, B., Limp, F.: Evaluation of community decision support software. National Consortium for Rural Geospatial Innovations in America. University of Wisconsin, Madison, WI. www.ruralgis.org, see Publications (2004)
32. Ormsby, T., Napoleon, E., Burke, R., Groessl, C., Feaster, L.: Getting to Know ArcGIS Desktop. 2nd edn. Updated for ArcGIS 9. ESRI Press (2004)
33. Maidment, D.R., Robayo, O., Merwade, V.: Hydrologic modeling. In: Maguire, D.J., Batty, M., Goodchild, M.F. (eds.) GIS, Spatial Analysis, and Modeling, pp. 319–332. ESRI Press (2005)
34. Batty, M.: Planning support systems: technologies that are driving planning. In: Geertman, S., Stillwell, J. (eds.) Planning Support Systems in Practice, pp. v-viii. Springer, New York, NY (2003)
35. Klosterman, R.E., Pettit C.J.: Guest editorial: an update on planning support systems. Environment and Planning B: Planning and Design **32**:477–484 (2005) London
36. For example, two such conferences include the Eighth International Conference on Computers in Urban Planning and Urban Management cited in [35] and held in Sendai, Japan, in 2003; and the Third Annual Public Participation Geographic Information Science (PPGIS) Conference held in Madison, WI, in 2004; the latter was co-sponsored by the Urban and Regional Information Systems Association (URISA) and the National Consortium for Rural Geospatial Innovations (www.ruralgis.org) – this conference was included in the citation for [13]
37. Geertman, S., Stillwell, J. (eds.): Planning Support Systems in Practice. Springer, New York, NY (2003)
38. Brail, R.K., Klosterman, R.E. (eds.): Planning Support Systems: integrating geographic information systems, models, and visualization tools. ESRI Press (2001)
39. U.S. EPA: Projecting Land-Use Change: A Summary of Models for Assessing the Effects of Community Growth and Change on Land-Use Patterns. EPA/600/R-00/098. U.S. Environmental Protection Agency, Office of Research and Development, Cincinnati, OH (2000)
40. The Orton Family Foundation: CommunityViz, Version 1.3. Course Training Manual. http://www.communityviz.com (2002)
41. Kwartler, M., Bernard, R.N.: CommunityViz: an integrated planning support system. In: Brail, R.K., Klosterman, R.E. (eds.) Planning Support Systems: integrating geographic information systems, models, and visualization tools, pp. 285–308. ESRI Press (2001) see also http://www.communityviz.com
42. McColl, C., Aggett, G.: Boom or Bust? A Case Study of PSS & PPGIS Implemented within a Rural Setting. In: Proceedings of the Annual PPGIS Conference of the Urban and Regional Information Systems Association (URISA), pp. 145–157. University of Wisconsin, Madison, WI (2004)
43. Aggett, G., McColl, C.: Breaking down barriers: geospatial decision-support systems help citizens embrace community land-use planning. Land Bulletin of the National Consortium for Rural Geospatial Innovations. University of Wisconson, Madison, WI (2006) www.ruralgis.org
44. Klosterman, R.E.: The What if? planning support system. In: Brail, R.K., Klosterman, R.E. (eds.) Planning Support Systems: Integrating geographic information systems, models, and visualization tools, pp. 263–284. ESRI Press (2001) See also http://www.what-if-pss.com
45. Klosterman, R.E.: (1999) The What If? Collaborative Planning Support System. Environment and Planning B: Planning and Design **26**:398–408 (1999) See also http://www.what-if-pss.com
46. Klosterman, R.E., Siebert, L., Hoque, M.A., Kim, J., Parveen, A.: Using an operational planning support system to evaluate farmland preservation policies. In: Geertman, S., Stillwell, J. (eds.) Planning Support Systems in Practice, pp. 391–407. Springer, New York, NY (2003) see also What if? at http://www.what-if-pss.com
47. Pettit, C.J.: Use of a collaborative GIS-based planning-support system to assist in formulating a sustainable-development scenario for Hervey Bay, Australia. Environment and Planning B: Planning and Design **32**:523–545 (2005)
48. Gaunt, C., Jackson, L.: Models for assessing the effects of community change on land use patterns. In: Geertman, S., Stillwell, J. (eds.) Planning Support Systems in Practice, pp. 351–371. Springer, New York, NY (2003)
49. Carsjens, G.J., van Lammeren, R.J.A., Ligtenberg, A.: STEPP: a strategic tool for integrating environmental aspects into planning procedures. In: Geertman, S., Stillwell, J. (eds.) Planning Support Systems in Practice, pp. 139–154. Springer, New York, NY (2003)
50. Ventura, S., Miller, T., Barry, G.: Community engagement in land use planning through web-based technologies. In: Geertman, S., Stillwell, J. (eds.) Planning Support Systems in Practice, pp. 87–98. Springer, New York, NY (2003) See also web site for study at http://www.lic.wisc.edu/shapingdane/
51. Pettit, C., Shyy, T., Stinson, R.: An on-line planning support system to evaluate urban and regional planning scenarios. In: Geertman S., Stillwell, J. (eds.) Planning Support Systems in Practice, pp. 331–347. Springer, New York, NY (2003)

52. U.S. EPA: National Center for Environmental Assessment. http://www.epa.gov/nheerl/wtc/index.html. WTC (World Trade Center) Report Link: Exposure and Human Health Evaluation of Airborne Pollution from the World Trade Center Disaster (External Review Draft). WTC Report: Toxicological Effects of Fine Particulate Matter Derived from the Destruction of the World Trade Center. (2002)
53. U.S. EPA: Strategic Plan for Homeland Security. http://www.epa.gov/epahome/downloads/epa_homeland_security_strategic_plan.pdf (2002)
54. Schwartz, P., Randall, D.: An Abrupt Climate Change Scenario and Its Implications for United States National Security. Report to the Pentagon. www.mindfully.org/Air/2003/Pentagon-Climate-Change1oct03.htm. (accessed December 23, 2006) (2003)
55. Writing of aspects of this paper was funded and supported by the USDA (Cooperative State Research, Education, and Extension Service) Rural GIS program (see www.ruralgis.org) and EPA's Office of Environmental Information

Environmental Planning Support Systems

► Environmental Planning and Simulation Tools

Environmental Sensor Networks

► Geosensor Networks

Environment-Sensitive Access Control

► Security Models, Geospatial

Epidemics

► Pandemics, Detection and Management

Epidemiological Mapping

► Bioinformatics, Spatial Aspects

Epidemiology, Computational

► Pandemics, Detection and Management

Epidemiology, Landscape

► Exploratory Spatial Analysis in Disease Ecology

Epidemiology, Spatial

► Exploratory Spatial Analysis in Disease Ecology

Error

► Uncertain Environmental Variables in GIS

Error Propagation

► Imprecision and Spatial Uncertainty

Error Propagation in Spatial Prediction

ROGER BIVAND
Economic Geography Section, Department of Economics, Norwegian School of Economics and Business Administration, Bergen, Norway

Synonyms

Change of support problem; COSP; Modifiable areal unit problem; MAUP

Definition

Spatial prediction involves the use of observations at given locations to make predictions of quantities of interest at locations for which observations on the variable of interest are not available, using a chosen method. If the locations and all variables are measured exactly, then the prediction errors are limited to those related to the choice of prediction method and the fitting of its parameters.

If, on the other hand, there is measurement error in the position and/or attributes, the output prediction errors will additionally be impacted by the uncertainty associated with the input positions and/or attribute values. This is what is known as the error propagation: uncertainty cascades through a sequence of operations on spatial data. When uncertainty propagates in this way, both prediction point estimates and distributions may be affected.

For simplicity, spatial positions are assumed here to be measured without error, so that error stems from attribute measurement error, choice of prediction method, and prediction method parameter fitting. Contemporary GIS do not handle positional measurement error adequately, and this discussion will be restricted to what is termed thematic error by Leung et al. (2004, p. 326). A comprehensive discussion is provided by Heuvelink (1998), who stresses the importance of the cascading of errors through GIS operations, in particular resampling and other change of

Error Propagation in Spatial Prediction, Table 1 Changes of support Gotway and Young (2002, p. 634)

We observe or analyze	But the nature of the process is	Examples
Point	Point	Point kriging; prediction of undersampled variables
Area	Point	Ecological inference; quadrat counts
Point	Line	Contouring
Point	Area	Use of areal centroids; spatial smoothing; block kriging
Area	Area	The modifiable areal unit problem (MAUP); areal interpolation; incompatible or misaligned zones
Point	Surface	Trend surface analysis; environmental monitoring; exposure assessment
Area	Surface	Remote sensing; multiresolution images; image analysis

support operations; another view is provided by Zhang and Goodchild (2002).

Change of support, whether implicit or explicit, potentially induces error in attribute measurements by transforming the original data to a new spatial representation. Arbia (1989) names the situation in which the spatial process of interest is of one form but the data observed are of another *spatial data transformation*. The form or support of the data is described by Burrough and McDonnell (1998, p. 101–102) as the area or volume of the physical sample on which a measurement is made. Gotway and Young (2002, p. 634) define support more completely as the size or volume associated with each data value, including the geometrical size, shape, and spatial orientation of the regions associated with the measurements. Table 1 tabulates an interesting set of examples which often occur in practice, in which our operationalizations of observations differ in form (support) from the underlying data generation process. The modifiable areal unit problem (MAUP) is a special case where boundaries for aggregates can be modified arbitrarily, such as voting districts (cf. gerrymandering) or school districts—changes in the boundaries of units can change the results of analysis substantially.

Error propagation involves the cascading of the consequences of choices and transformations affecting the data of interest. Even when spatial position is taken as measured without error, attributes are subject to uncertainty, uncertainty that is impacted by data transformations, and it is important to acknowledge that this uncertainty is also information of value.

Historical Background

In some application areas, the "fixing" of transformed data has only restricted consequences. Deriving slope and aspect from a digital elevation model (DEM), all with specific support, and then querying those layers at the point location of a house on sale may give a reasonable idea of conditions at the house. Usually, field surveying would show that the resolution and implementation of the DEM affect the elevation, and the derived slope and aspect will depend on the DEM and the implementation of the slope/aspect operation. So one ought not to be surprised by the induced uncertainty, that elevation has a point estimate and a distribution within which the point estimate lies, with the same applying to slope and aspect. It is not likely that a house purchaser would sue the seller or her agent for getting the aspect wrong, for stating west-facing when in fact north-west would be more accurate.

Burrough and McDonnell (1998, p. 102) present the case of soil samples in discussion of the classic Meuse bank soil pollution data set. The soil samples were taken from sites chosen, but are not point locations. Soil samples here were collected from 10 m by 10 m areas using a stratified random sampling scheme and bulked up, including depths from 0 cm to 20 cm. These bulked samples were then analyzed in the laboratory, and measurements derived for a number of attributes. The values reported are "good practice", but have clearly smoothed the spatial variation in attribute values, such as heavy metal pollution. There are pragmatic grounds for the smoothing, but the measurements themselves are a representation of reality, not reality. As Gotway and Young (2002) point out, the consequences of change of support in other application areas are much more severe. In the analysis of public health data, it is typically the case that information is available on health outcomes, often with point or aggregated area measures of incidence. Importantly, observations of environmental factors possibly related to illness are only very rarely available with the same temporal and spatial support. Air quality monitoring, or aerobiological (pollen) monitoring data are typically available for points, and need to be interpolated in order to reach an estimate of patient exposure. However, most commonly used interpolation methods do not incorporate knowledge of air mass movement. As Guttorp (2000) comments: "Studies using personal monitors may be helpful in order to assess more precisely the health effects of a given exposure. Current technology, however, produces rather unwieldy monitors, which are likely to affect personal behavior." Guttorp (2003) gives a broad overview of the challenges involved in using spatially transformed environmental data when measurement and interpolation uncertainty cascades through to the real relationships of interest.

Finally, both representational operationalizations and algorithm implementations can lead to uncertainty in attribute data, uncertainty that ought to be acknowledged in "downstream" use of the attributes. In an interesting comparison of viewshed (line-of-sight) implementations in a range of GIS software, Fisher (1993) found large differences even when the approaches described in documentation appeared superficially to be the same. A derived attribute from the same DEM, the same point locations, and other arguments to the GIS operations could differ for the same target location depending on which software implementation was chosen. In another critical paper, Fisher (1997) points up the conceptual weaknesses of the pixel as a representation of reality. Analytical, chiefly statistical, methods are required, in order to carry through the uncertainty associated with raster cells.

Scientific Fundamentals

Error propagation and the change of support problem has been described here so far with reference to Heuvelink (1998) and Gotway and Young (2002). Mathematical statistics has been concerned with these problems for over half a century, starting among others from Kolmogorov (1941) and also Gandin (1963), who was working on applications in meteorology. The background to contemporary geostatistics and more generally spatial statistics is quite complex, with different scientific disciplines, typically with quite different measurement contexts, and often very different spatial scales—meteorology and precision agriculture, for example—developing in partial isolation from each other. Most of the status of spatial statistics is covered in detail in Cressie (1993), and brought up to the present by Schabenberger and Gotway (2005).

Because of the size, complexity, and considerable differences in approaches and notations covering error propagation and the change of support problem, it seems best to refer the reader to the standard works. It is also helpful to turn to examples, such as that given by Waller and Gotway (2004, p. 311–318) of kriging areal regions, that is making predictions from point support data to areal support. Since these topics are as inaccessible as they are, this example showing how some of these methods may be carried out using software interfacing with GIS can show practical ways of catering for change of support. Banerjee, Carlin and Gelfand (2004, p. 175–215) cover what they term spatial misalignment in a useful chapter with examples. The examples cover both hierarchical Bayesian methods for interpolation and smoothing of Poisson responses with covariates in the nested case, and hierarchical Bayesian interpolation and smoothing over non-nested misaligned data grids.

At this point, it should be clear that Wikle (2003, p. 196) is right to stress the need for continuous interaction between domain scientists and statisticians at every stage of research in which change of support or error propagation may be encountered. What is important is to be able to design the data collection of the research project in such a way as to make it possible to "translate this scientific knowledge into reasonable statistical models", thus permitting uncertainty to be handled. A range of methods are becoming available for fitting hierarchical models, with tiers of cascading uncertainty, both using fully Bayesian approaches, and by simulating from fitted models, for example parametric bootstrapping in simpler cases. Gelfand et al. (2001) and Gneiting and Schlather (2002) tackle the challenging topic of handling spatio-temporal error processes; this is a difficult but important area, as is the adequate handling of three-dimensional processes. Contemporary advances in geostatistics are demonstrated by Diggle and Ribeiro (2007), who like many of the other active applied statisticians have provided software implementations of their theoretical findings. In fact it is now difficult to imagine useful results in this area without software implementations, not least because almost no analytical results are available, and almost all statisticians end up using simulation methods to represent the cascading of uncertainty through to the findings of interest.

Key Applications

At present, there are few examples of the actual use of error propagation handling techniques in GIS software outside academia. In general it has proved difficult even to accommodate positional accuracy information in GIS in more than metadata form, let alone attribute uncertainty. Fortunately, most implementations of geostatistical methods interfaced with GIS do provide estimates of prediction errors, but it is not, to put it mildly, always the case that users inspect this information with any care. Consequently, handling error propagation in spatial prediction involves the interfacing of GIS to other software systems, and fundamentally challenges the data models used in GIS to contain crisp (Burrough 1996) and fully known data.

Future Directions

Since so little is available off the shelf at present, future developments have to hold promise. They will naturally involve the recommendations made by Wikle above, that is the need for statisticians, professionals in uncertainty handling, to be involved in serious studies from an early point, certainly before any data collection is finalized. Krivoruchko and Gotway (2003, p. 2) point up the value of a probabilistic view of geoprocessing: "our uncertainty about reality and the impact of our assumptions and estimates on conclusions and decisions can be integrated with the model using a probabilistic framework. We can use the probabilistic interpretation to provide information that can be extremely useful for decision-making."

The academic literature is replete with work providing fresh avenues for making this possible. The monographs by Rue and Held (2005) and Diggle and Ribeiro (2007) show how simulation and model-based methods may be used to handle uncertainty in more productive and useful ways. The same applies to recent papers by Barry (2005), Cornford et al. (2005), Fuentes and Raftery (2005) and Pardo-Igúzquiza et al. (2006), all increasing insight into ways of using uncertainty creatively. Fuentes and Raftery (2005) show how the output of numerical models can be integrated with observations using Bayesian methods, a potentially very fruitful area given the importance of modeling in expressing and representing underlying scientific knowledge. These are just a few of a large and increasing range of work trying to draw useful information from uncertainty cascading through hierarchical levels of a research process, so a conclusion that error propagation will be better accommodated in the future than at present is not unduly rash.

Cross References

► Decision Support Systems

Recommended Reading

Arbia, G.: Statistical effect of data transformations: a proposed general framework. In: Goodchild, M.F., Gopal, S. (eds.) The accuracy of spatial databases, 249–259. Taylor & Francis, London (1989)

Banerjee S., Carlin BP., Gelfand A.E.: Hierarchical modeling and analysis for spatial data. Chapman & Hall/CRC, Boca Raton (2004)

Barry, R.P.: The local approximation of variograms in R^2 by covariance functions. Stat. Probab. Lett. **74**, 171–177 (2005)

Burrough, P.A.: Natural objects with indeterminate boundaries. In: Burrough, P.A., Frank, A.U. (eds.) Geographic objects with indeterminate boundaries, 3–28. Taylor & Francis, London (1996)

Burrough, P.A., McDonnell, R.A.: Principles of geographical information systems. Oxford University Press, Oxford (1998)

Cornford, D., Csató, L., Opper, M.: Sequential, Bayesian geostatistics: a principled method for large data sets. Geogr. Anal. **37**, 183–199 (2005)

Cressie, N.A.C.: Statistics for spatial data. Wiley, New York (1993)

Diggle, P.J., Ribeiro, P.J.: Model-based Geostatistics. Springer, New York (2007)

Fisher, P.F.: Algorithm and implementation uncertainty in viewshed analysis. Int. J. Geogr. Inf. Syst. **7**, 331–347 (1993)

Fisher, P.F.: The pixel: a snare and a delusion. Int. J. Remote Sens. **18**, 679–685 (1997)

Fuentes, M., Raftery, A.E.: Model evaluation and spatial interpolation by Bayesian combination of observations with outputs from numerical models. Biom. **61**, 36–45 (2005)

Gandin, L.S.: Objective Analysis of Meteorological Fields. Gidrometeorologicheskoe Izdatel'stvo (GIMIZ), Leningrad (1963) (translated by Israel Program for Scientific Translations, Jerusalem, 1965)

Gelfand, A.E., Zhu, L., Carlin, B.P.: On the change of support problem for spatio-temporal data. Biostat. **2**, 31–45 (2001)

Gneiting, T., Schlather, M.: Space-time covariance models. In: El-Shaarawi, A.H., Piegorsch, W.W. (eds.) Encyclopedia of Environmetrics, Wiley, Chichester, 2041–2045 (2002)

Gotway, C.A., Young, L.J.: Combining incompatible spatial data. J. Am. Stat. Assoc. **97**, 632–648 (2002)

Guttorp, P.: Environmental statistics—a personal view. Int. Stat. Review **71**, 169–179 (2003)

Guttorp, P.: Setting Environmental Standards: A Statistician's Perspective. NRCSE Technical Report 048, http://www.nrcse.washington.edu/pdf/trs48_setting.pdf (2000)

Heuvelink, G.B.M.: Error propagation in environmental modelling with GIS. Taylor & Francis, London (1998)

Kolmogorov, A.N.: Interpolation and extrapolation of stationary random sequences. Isvestiia Akademii Nauk SSSR, Seriia Matematicheskiia **5**, 3–14 (1941) (Translation, 1946, Memo RM-3090-PR, Rand Corp., Santa Monica, CA)

Krivoruchko, K., Gotway, C.A.: Assessing the Uncertainty Resulting from Geoprocessing Operations. submitted for publication, ESRI Press, http://www.esri.com/library/whitepapers/pdfs/assessing-uncertainty.pdf (2003)

Leung, Y., Ma, J.-H., Goodchild, M.F.: A general framework for error analysis in measurement-based GIS I: the basic measurement-error model and related concepts. J. Geogr. Syst. **6**, 325–354 (2004)

Pardo-Igúzquiza, E., Chica-Olmo, M., Atkinson, P.M.: Downscaling cokriging for image sharpening. Remote Sens. Env. **102**, 86–98 (2006)

Rue, H., Held, L.: Gaussian Markov random fields. Chapman & Hall/CRC, Boca Raton (2005)

Schabenberger, O.:, Gotway, C.A.: Statistical Methods for Spatial Data Analysis. Chapman & Hall/CRC, Boca Raton (2005)

Waller, L.A., Gotway, C.A. Applied spatial statistics for public health data. Wiley, Hoboken, N.J. (2004)

Wikle, C.K.: Hierarchical models in environmental science. Int. Stat. Review **71**, 181–199 (2003)

Zhang, J., Goodchild, M.F.: Uncertainty in geographical information. Taylor & Francis, London (2002)

E-Science

► Cyberinfrastructure for Spatial Data Integration

ESRI

► ArcGIS: General Purpose GIS Software System

Estimation, Non-Parametric

▶ Geosensor Networks, Estimating Continuous Phenonena

Estimation, Parametric

▶ Geosensor Networks, Estimating Continuous Phenonena

Estimation Predication

▶ Geosensor Networks, Estimating Continuous Phenonena

Euclidean Distance

▶ Distance Metrics

Euclidean Restriction

▶ Nearest Neighbor Queries in Network Databases

Euler's Konigsberg's Bridges Problem

▶ Graph Theory, Konigsberg Problem

Evacuation Logistics Planning

▶ Emergency Evacuation Plan Maintenance

Evacuation Model Systems

▶ Emergency Evacuations, Transportation Networks

Evacuation Planning

▶ Contraflow for Evacuation Traffic Management

Evacuation Routes

▶ Contraflow in Transportation Network

Event

▶ Temporal GIS and Applications

Event, Cyclic

▶ Modeling Cycles in Geospatial Domains

Event, Periodic

▶ Modeling Cycles in Geospatial Domains

Event, Recurrent

▶ Modeling Cycles in Geospatial Domains

Evidence

▶ Computing Fitness of Use of Geospatial Datasets

Evolution

▶ Indexing Spatio-temporal Archives

Evolution, Landscape

▶ Geographic Dynamics, Visualization And Modeling

Evolution of Earth Observation

DANIEL J. SARLITTO
SummitPoint Management LLC, Issaquah, WA, USA

Synonyms

Digital image processing; Geospatial technology; Earth observation; LiDAR; RADAR; GPS; Surveillance; Remote sensing, satellite-based; Remote sensing, aerial; Intertial motion unit (IMU)

Definition

Remote Sensing dates back to ancient Napoleonic times when the French first employed hot-air balloons to assess "from above" enemy force and position prior to battle. Today, although not much has changed in terms of the mainstay application, military surveillance, certainly much has evolved in the acquiring and processing technologies, applications, and audience for image data. Earth Observation, in the context of this article, refers to the indirect, passive observation or "remote sensing" of the Earth from above. Remote sensors, satellite- and aerial-based platforms, usually record Electromagnetic Radiation (EMR) from a target source, essentially creating a high-speed communications link between a sensor and a target being sensed. The information that can be derived from a target's EMR reflectance characteristics, inside and outside the visible spectrum, provide the interpreter invaluable insight about the target. When combined with active remote technologies, such as the Global Positioning System (GPS), Light Detection and Ranging (LiDAR), and Radio Detection and Ranging (RADAR), the absolute *x*/*y* location and height (*z*) of the target can be determined as well, all with extraordinary precision and accuracy. Furthermore, much of the high resolution image and location data that was once only available to the defense community for military surveillance applications is now available on the open market for a host of public and private, consumer and professional applications. The following article provides a brief overview of earth observation in terms of the evolution of data acquisition, processing and distribution.

Historical Background

Since its start in the early 1960s, satellite-based remote sensing has been a large government operation primarily with military and meteorological goals and objectives. The inherently global, synoptic view afforded by a satellite platform, and the substantial cost associated with the design and construction of a satellite and launch vehicle, are the main reasons why these were, and continue to be, government-underwritten programs. Yet, although the majority of the technology developed in the early years was driven by military goals associated with the Cold War, there were, and continue to be, "friendly" application drivers as well, such as the remote detection of "biophysical" variables. Remotely sensed data gathered from specific EMR wavelengths can provide direct biological and/or physical information about the target. Vegetation, water, soils and rocks, atmosphere, and urban structure are examples of some biophysical variables measurable from space [3].

The United States' Landsat initiative (originally called the Earth Resources Technology Satellite or ERTS-1) is the oldest civilian satellite-based remote sensing program with the first optical satellite (Landsat 1) launched in the early 1970s, and the last, Landsat 7, launched just prior to the millennium in 1999. Landsat 7's Enhanced Thematic Mapper (ETM) sensor, with its range of spectral sensitivity and large swath, makes it particularly well-suited for biophysical and environmental assessment applications. Therefore, Landsat continues to be the preferred source data for many scientific applications, such as wetlands analysis, determination of vegetative health, etc. The former Soviet Union had quite a history of satellite-based "spy" surveillance dating back to the mid-1960s; 30 years later Russia was the first country to enter the high resolution commercial imagery business when it began releasing scanned KVR-1000 data (no better than 2 meter) to the open international market. Interestingly, shortly after Russia's release of KVR-1000 data, the US high-resolution commercial satellite programs began to emerge including IKONOS (1meter), QuickBird (sub-meter), and OrbView (sub-meter). France's Centre National d'Études Spatiales (CNES) remote sensing program also dates back to the early 1970s; France launched its first satellite (Satellite Pour l'Observation de la Terre or SPOT-1) in 1986. It is worth noting that SPOT employed an innovative high-resolution visible (HRV) charge-coupled device (CCD) sensor, and the program as a whole represents a successful government-private sector venture. Japan's initial foray into remote sensing took a very similar approach to that of the US's Landsat program. That country's system design goals were primarily defined to meet the need of research and public-purposed applications; Japan launched its first Marine Observation Satellite (MOS) in 1987. Since then, Japan has taken a multi-purpose design perspective with its Advance Land Observing Satellites (ALOS) and Information Gathering Satellites (IGS). Additionally, India launched its first Indian Remote Sensing Satellite (IRS) in 1988 and Canada launched RADARSAT-1 in 1995 [2]. All of the above countries (US, Russia, France, Japan, India, and Canada) represent "early-adopter" satellite-based remote sensing programs that have evolved into very robust and mature programs. And despite the commercialization effort that began in the early 1980s, current and planned satellite programs continue to be financed directly or indirectly by individual governments or government-consortiums; for example, the European Space Agency (ESA) and Disaster Management Constellation (DMC).

Figure 1 shows current and planned land imaging satellites and their host nations. It is worth noting that of the approximately 2400 Earth Observing satellites in orbit, less than five percent are commercial [4], and the current community of nations is no longer limited to a few. Today, government and non-government organizations are combining resources and sharing expertise in a more collaborative, market-driven approach to program management, and many developing countries have emerging programs, some on a shoestring budget, as well.

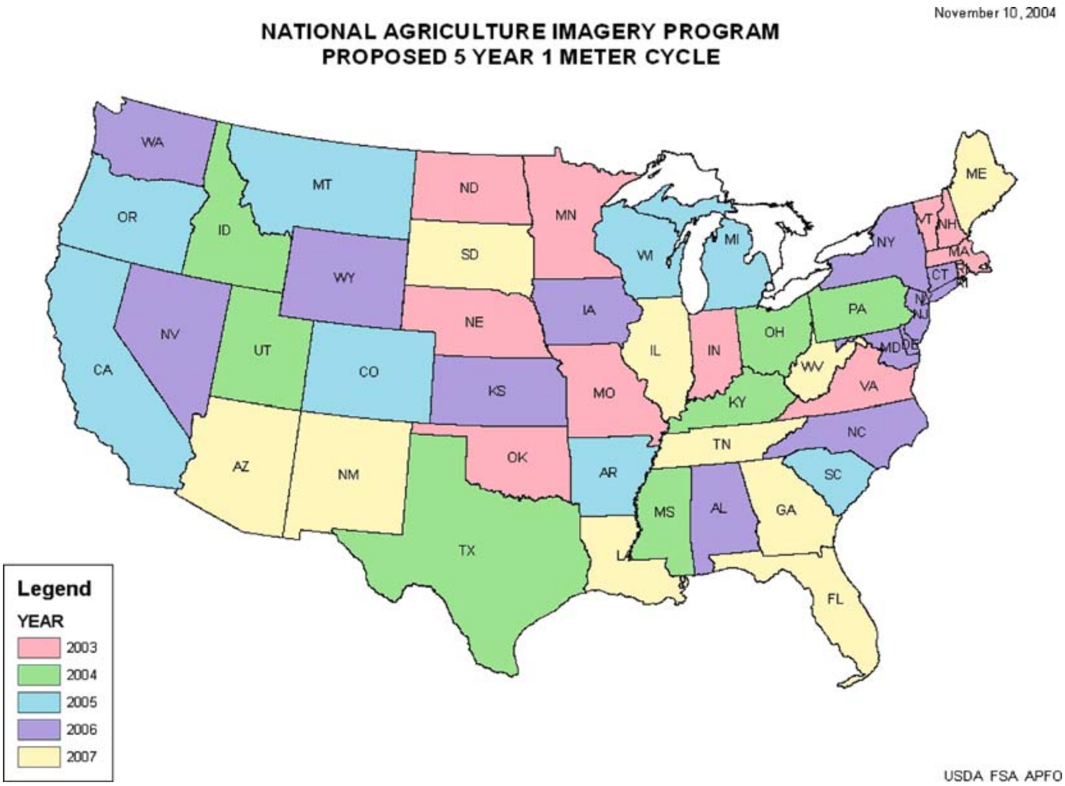
CURRENT AND PLANNED, 36 M & BETTER, LAND IMAGING SATELLITES

SATELLITE	COUNTRY	LAUNCH	PAN RES. M	MS RES. M	SWATH KM
	OPTICAL				
DMC AlSat-1(SSTL)	Algeria	11/28/02		32	600
Tsinghua-1 (SSTL)	China	06/28/00		39	600
DMC China DMC	China	10/27/05	4.0	32	600
CBERS-2	China/Brazil	10/21/03	20.0	20	113
CBERS-2B	China/Brazil	01/15/06	20.0	20	113
CBERS-3	China/Brazil	05/01/08	5.0	20	60, 120
CBERS-4	China/Brazil	06/01/10	5.0	20	60, 120
Proba	ESA	10/22/01	8.0	18, 36	14
SPOT-2	France	01/22/90	10.0	20	120
SPOT-4	France	03/24/98	10.0	20	120
SPOT-5	France	05/04/02	2.5	10	120
Pleiades-1	France	07/01/08	0.7	2.8	20
Pleiades-2	France	07/01/09	0.7	2.8	20
RapidEye-A	*Germany**	06/01/07		6.5	78
RapidEye-B	*Germany**	06/01/07		6.5	78
RapidEye-C	*Germany**	06/01/07		6.5	78
RapidEye-D	*Germany**	06/01/07		6.5	78
RapidEye-E	*Germany**	06/01/07		6.5	78
IRS 1C	India	12/28/95	6.0	23	70, 142
IRS 1D	India	09/29/97	6.0	23	70, 142
IRS ResourceSat-1	India	10/17/03	6.0	6, 23, 56	24, 140,740
IRS Cartosat 1	India	05/04/05	2.5		30
IRS Cartosat 2	India	12/10/05	1.0		10
IRS ResourceSat-2	India	01/15/06	6.0	6, 23, 56	24, 140, 740
EROS A1	*Israel**	12/05/00	1.8		14
EROS B1	*Israel**	12/05/05	0.7		7
EROS C	*Israel**	03/01/08	0.7	2.5	16
TERRA (ASTER)	Japan/US	12/15/99		15, 30, 90	60
ALOS	Japan	07/01/06	2.5	10	35, 70
KOMPSAT-1	Korea	12/20/99	6.6		17
KOMPSAT-2	Korea	11/20/05	1.0	4	15
RazakSat	Malyasia	02/01/06	2.5	5	?
DMC NigeriaSat-1 (SSTL)	Nigeria	09/27/03		32	600
MONITOR-E -1	Russia	08/26/05	8.0	20	94, 160
Resurs DK-1 (01-N5)	Russia	10/31/05	1.0	3	28
X-Sat	Singapore	01/15/06		10	50
R26m	South Africa	09/01/06		7.5	?
RocSat2	Taiwan	04/20/04	2.0	8	24
DMC ThaiPhat (SSTL)	Thailand	12/01/04		36	600
THOES	Thailand	06/30/07	2.0	15	22,90
DMC BilSat (SSTL)	Turkey	09/27/03	12.0	26	52
DMC UK (SSTL)	UK	09/27/03		32	600
TopSat (SSTL)	UK	10/27/05	2.5	5	10, 15
Landsat 5	US	03/01/84		30.0	185
Landsat 7	US	04/15/99	15.0	30	185
IKONOS-2	*US**	09/24/99	1.0	4	11
MTI	US	03/12/00		5, 20	12
EO-1	US	12/07/00	10.0	30	37
QuickBird-2	*US**	10/18/01	0.6	2.5	16
OrbView 3	*US**	06/26/03	1.0	4	8
OrbView5	*US**	03/16/07	0.41	1.64	15
WorldView -1	*US**	11/11/05	0.5		16
WorldView -2	*US**	07/01/08	0.5	1.8	16
LDCM (NPOES)	US	06/30/09	10.0	30	177
DMC VinSat-1	Vietnam	05/01/06	4.0	32	600
	RADAR				
RadarSat 1	Canada	11/04/95	8.5		
RadarSat 2	Canada	01/15/06	3.0		
ERS-2	ESA	04/21/95	30.0		
ENVISAT	ESA	03/01/02	30.0		
TerraSAR X	Germany	04/15/06	1.0		
TerraSAR L	Germany	06/15/08	1.0		
RISAT	India	01/30/07	3.0		
COSMO-Skymed-1	Italy	02/01/06	1.0		
COSMO-Skymed-2	Italy	08/01/06	1.0		
COSMO-Skymed-3	Italy	02/01/07	1.0		
COSMO-Skymed-4	Italy	10/01/07	1.0		
ALOS	Japan	07/01/06	10.0		

Revised 10/30/05

Evolution of Earth Observation, Figure 1

Unlike satellite-based remote sensing, aerial programs (at least in the US) have largely been federally-contracted civilian operations with the traditional technology used being film – black and white, color, and color infrared (false-color) photography. Much of the photography acquired in the US since the early 1930's has been used to produce the United States Geological Survey's (USGS) topographic map series, still in widespread use



Evolution of Earth Observation, Figure 2

today, and by a number of agencies including the US Forest Service, Bureau of Reclamation, Agriculture, and others, for soils, mining, forest, and a range of land-use/land-cover/transportation mapping applications. The National High Altitude Photography (NHAP) program was established in 1978 to consolidate the many disparate federal aerial photography programs and create a single, contiguous, and regularly updated US coverage to serve the needs of all public agencies. This program generated a huge archive of photography. NAPP (National Aerial Photography Program) was created in 1986 to coordinate the continued collection and distribution of aerial photography, and the National Agriculture Imagery Program (NAIP), overseen by the US Department of Agriculture (USDA), is the current manifestation of the US's aerial program. Data acquired as part of the NAIP program is no longer limited to film-based aerial photography rather today's program specifies data products that leverage multiple remote sensing technologies. Figure 2, sourced from the USDA, shows the proposed 5-year 1-meter US coverage plan up to 2007. Emerging airborne systems are increasingly digital versus film-based, and collect more than just photography and/or imagery, they also collect topography. In terms of imagery, these newer systems acquire multi-spectral data using one of several commercially-available digital linear array sensor technologies with each sensor capturing a specific slice of the EMS – ultraviolet, infrared, thermal and microwave, in addition to visible at a specific orientation.

An airborne digital platform, with which the author has direct experience, utilizes a twin-engine, turboprop aircraft. This pressurized cabin aircraft is capable of airspeeds from 120 to 245 knots, has a ceiling altitude of 30,000 feet, and has been customized with equipment ports to accept a digital linear array camera for collecting imagery, as well as a LiDAR system for collecting elevation data. This camera/LiDAR configuration is among the most sophisticated airborne platforms available as of this writing. The camera uses ten sensors to simultaneously capture color, near-infrared and stereo panchromatic imagery. A dedicated IMU (Inertial Motion Unit) allows automated rectification of the imagery to account for roll, pitch, and yaw of the aircraft. In addition to the digital linear array camera, this platform includes an airborne laser scanner. This state of the art LiDAR system has a field of view of up to 75 (d), mirror oscillations as precise as 0.001(d), and a 15–30 kHz laser pulse rate. The laser configuration also contains a dedicated IMU (separate from the camera's IMU) for automated rectification of raw elevation data. In addition to this hardware, the laser system is accompanied by advanced software to aid in the post processing of the data.

Key Applications

But aerial also has its disadvantages, such as turnaround time, limited swath or small coverage area, and logistical constraints including airspace restrictions, while the main disadvantage of satellite-based imaging continues to be its comparatively lower spatial resolution, inability to penetrate cloud cover, high cost and in some cases, government control. Consequently, for large-area applications (US statewide coverage, for instance) it is common to obtain satellite imagery for the area of interest to support mission planning of a pending airborne collection. On the other hand, for applications where the resolution and quality of off-the-shelf satellite imagery is satisfactory, it is common to procure satellite imagery for the full area then simply fill in the gaps with aerial data; coverage gaps in satellite imagery are often associated with atmospheric conditions. However, it is becoming increasingly more common to fill aerial coverage gaps with high-resolution commercial satellite imagery as opposed to the other way around.

So from a technology standpoint, current aerial and satellite-based remote sensing platforms complement more than they compete with each other. Airborne systems can provide data on demand, operate in adverse weather conditions (flying under clouds), have adaptable resolution (ranging from 0.1–0.8 m panchromatic and 0.2–1.6 m multi-spectral) by changing flying height, and stereo imagery is inherent to the platform. Satellite systems, on the other hand, operate at a fixed orbit (450–650 km), data availability is weather dependent (cloud cover is the most limiting parameter in satellite data), they operate at a fixed resolution (ranging from 0.8 m panchromatic to 4.0 m multi-spectral), and support stereo on demand.

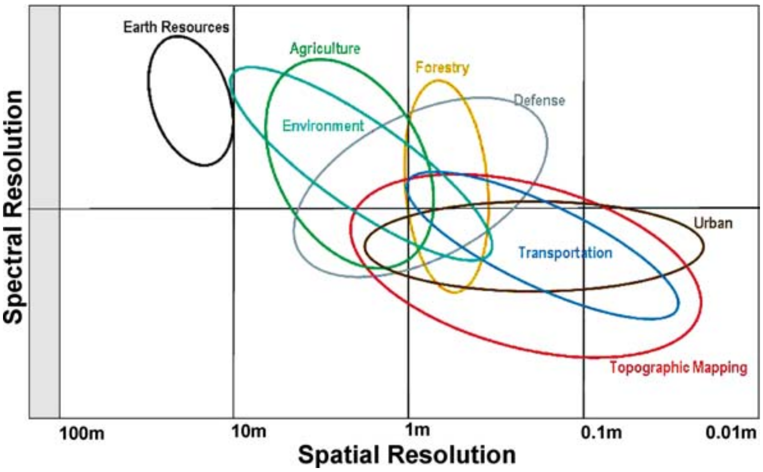
Yet, from a revenue standpoint, excluding military expenditure, aerial still exceeds satellite primarily because the platform, on-board technology, and logistics make it ideal for the majority of friendly remote sensing applications, which are local, human-scale, applications. Public and/or private Engineering and Geographic Information System (GIS)-based mapping applications including transportation planning, natural resource management (forestry, mining and energy), urban planning, and agriculture and soils, to name a few. Even today, the high resolution, selective and un-obstructed view potential, and revisit capability of an aircraft-mounted film-based camera or digital sensor is significantly greater than anything achievable from space.

Today, the gap between space and air is closing. In terms of acquisition, satellite and aerial platforms employ very similar digital sensor technologies (passive and active); and can collect thermal, multi- and hyper-spectral image data at very comparable spatial resolutions. In terms of processing and delivery, digital platforms (satellite and aerial) will soon be able to deliver data nearly on-demand as processing routines for image data, regardless of source, approach near full-automation. The figures below, sourced from Leica Geosystems LLC, depict common platform technologies and applications.

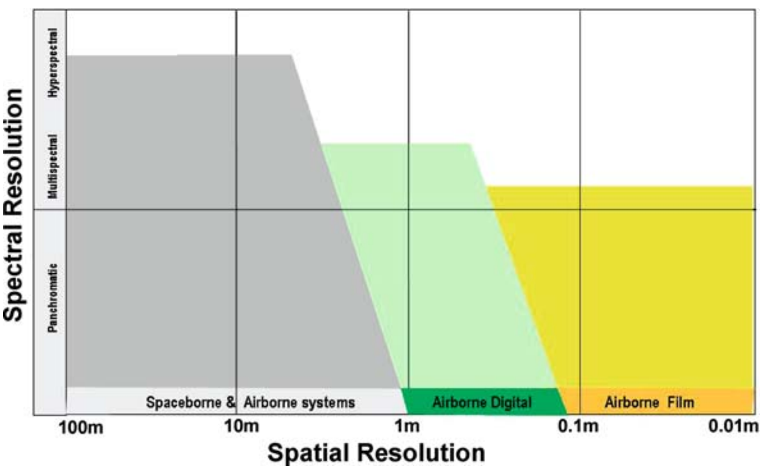
But the acquisition of digital data represents only a small part of the overall remote sensing process; raw data is pre-processed, enhanced, and post-processed utilizing a combination of sophisticated semi-automated software routines in order to create the final image products used in the full range of earth observation applications. Raw data is adjusted to compensate for known sensor characteristics (radiometric correction); distortions caused by the atmosphere must be removed (atmospheric correction); and data must be rectified/registered to a projection and grid system (geometric correction). The rectification/registration process generally involves the use of GPS-collected ground control points and orthorectification means that the imagery has been reconciled against an elevation dataset (Digital Elevation Model or DEM) and its horizontal geo-referencing is highly accurate. After pre-processing, a variety of image enhancements can be made depending on the intended audience and application. Examples of such image enhancements include compensation for natural variations in the imagery (color balancing), correction of slight horizontal discrepancies between adjacent images (mosaic), and automated adjustments to reduce visibility of seams (dithering), plus a host of standard and custom processes, such as filtering and classification (supervised and unsupervised). In general, imagery is pre-processed, enhanced, and post-processed in a standard, repeatable fashion in order to ensure temporal, spatial, and spectral compatibility between frames/scenes. And because data is processed in a standard way, and is organized in machine-friendly rows/columns, there continues to be less manual intervention; most routines have been automated such that the turnaround time, data collection to data delivery, has been reduced significantly.

Future Directions

In 2004, the American Society for Photogrammetry and Remote Sensing (ASPRS) estimated the Remote Sensing industry to be approximately $ 7 billion with an expected growth rate of 9 to 14 percent annually. ASPRS defines the industry as those commercial firms, not-for-profit orga-



Evolution of Earth Observation, Figure 3



Evolution of Earth Observation, Figure 4

nizations, government agencies, and academic institutions involved in the capture, production, distribution, and application of remotely sensed geospatial data and information. (Source: The American Society for Photogrammetry and Remote Sensing)

Government-sponsored commercial initiatives have stimulated this growth, yet, one unmistakable trend in earth observation points to high resolution remote sensing as a "business" with the predominant focus on higher spatial resolution and "pinpoint" location; spectral resolution has become a low priority. Consequently, as government satellites give way to commercial systems, the scientific community may have less data with which to work, which is an unfortunate trend.

Interestingly, however, the primary scientific application of remote sensing now has less to do with man's passive observation of the biosphere and more to do with the active detection of our impact on it; particularly the selective viewing and automatic detection of change via artificial intelligence. So the original applications of surveillance and weather have been joined by a variety of common international goals and objectives, such as tracking global climate change, draught, and deforestation to communicating "almost-live" world events via the news media. Another industry trend is the closing gap between existing platforms and the emergence of new ones. Today's satellite and airborne remote sensing systems employ very similar technologies, can collect data at very comparable resolutions, and can deliver data nearly on-demand as processing and delivery approaches full-automation. Yet, these two traditional platforms have been joined by a third – pedestrian-based data collection. Citizens everywhere are equipped with location-enabled picture phones, and are collecting images and sounds continuously. This logical convergence of technology has triggered an explosion of demand for virtual, searchable, content-rich data types of not just text and tables, but images and sounds, and companies are working to develop advanced search routines to supply this demand. The audience for these data includes consumers, businesses and government organizations with the venue for interaction a virtual online data super-store. Additionally, the accessibility to data is no longer tied to ownership or control of the acquiring technology, subject to considerable government restriction, or restricted to only those entities that can invest in considerable computer storage. In recent years, several private companies have built the requisite infrastructure to make data more accessible to a broader audience near-real-time via the internet. Advances in associated standards, such as ISO JPEG 2000 (JP2) and Geography Markup Language (GML), are integral to these market advances, as well as the value-added tools and technologies that are now being designed to facilitate true interoperability.

Clearly, there are geopolitical issues helping to drive the current business of earth observation. International tensions, especially since the 9/11 terrorist attacks on the World Trade Center, have certainly kept defense and security high on the priority list of most nations. Combine defense with natural disasters, such as the 2004 Sumatran Tsunami, and increased attention on global warming and there is an increased demand for four-dimensional high-resolution remote sensing of our environment. Furthermore, in addition to the traditional "remote" platforms – satellite and aerial – individual citizens have become another earth observing-platform generating real-time views of what exists at any given moment in any given place. So, the final trend in earth observation is this ubiquity of data; the push of timely, high-resolution data to any device, anytime, anywhere.

Cross References

- ► Bayesian Spatial Regression for Multi-source Predictive Mapping
- ► Intelligence, Geospatial
- ► Intergraph: Real Time Operational Geospatial Applications
- ► Photogrammetric Applications
- ► Standards, Critical Evaluation of Remote Sensing
- ► Temporal GIS and Applications

Recommended Reading

Avery, T.E., Berlin, G.L.: Interpretation of Aerial Photographs. 4th edn. Burgess Publishing, Minneapolis, MN (1985)

Baker, J.C., O'Connell, K.M., Williamson, R.A. (eds.): Commercial Observation Satellites: At the Leading Edge of Global Transparency. RAND-ASPRS, Santa Monica, CA (2001)

Jensen, J.R.: Remote Sensing of the Environment: An Earth Resource Perspective. Prentice-Hall, Upper Saddle River (2000)

Stoney, W.E.: Markets and Opportunities. Earth Imag. J. **2**(1), 10–14 (2005)

Evolution of GIS and LBS

► Computer Environments for GIS and CAD

Evolutionary Algorithms

► Geographic Dynamics, Visualization And Modeling

Evolving Spatial Patterns

► Patterns in Spatio-temporal Data

Exchange, Data

► OGC's Open Standards for Geospatial Interoperability

Exchange Format

► Geography Markup Language (GML)

Exploratory Cartography

► Exploratory Visualization

Exploratory Data Analysis

► Movement Patterns in Spatio-temporal Data

Exploratory Spatial Analysis

► Geographic Knowledge Discovery

Exploratory Spatial Analysis in Disease Ecology

LANCE A. WALLER
Department of Biostatistics, Rollins School of Public Health, Emory University, Atlanta, GA, USA

Synonyms

Conservation medicine; Epidemiology, spatial; Epidemiology, landscape

E

Definition

Disease ecology represents an intersection of population ecology, community ecology, epidemiology and environmental health focused on the study of (typically infectious) disease within a population of individuals. Analysis often focuses on the interactions of at-risk individuals with each other and with elements of their environment. Accordingly, spatial factors often play key roles in disease ecology due to the influence of heterogeneous geographic distributions (e. g., host distributions, vector abundance, and landscape barriers to contact) on the spread of an infectious disease.

The phrase "exploratory spatial analysis" encompasses a wide range of goals and techniques for investigating patterns and process within georeferenced data. The phrase intersects other phrases in the literature and merits brief discussion. Bailey and Gatrell (1995) distinguish between "spatial analysis" and "spatial data analysis" with the former referring to the exploration of spatial association through basic geographic information system operations such as layering, buffering, and linking databases via spatial location, and the latter referring to statistical analysis of spatially referenced observations. In addition, Haining (2003, Chapter 5) provides a detailed conceptual framework for "exploratory spatial data analysis" (ESDA), the primary goals of which are to summarize spatial properties and patterns and to formulate hypotheses from geographically referenced data using methods making minimal data assumptions and which are numerically and graphically resistant to the impact of isolated outlying observations. Such subtle distinctions have their place in the categorization of analytic techniques within geographic information science, but in the sections below, all of these components are stressed in the setting of disease ecology. More specifically, modern analysis of disease ecology often involves a synergistic interaction between geographic information systems, mathematical models of population dynamics, and statistical analysis of reported public health data. As a result, visual and quantitative summaries of complex spatial and spatio-temporal patterns play a key role in the study of disease ecology.

Historical Background

The end goal of most geographic analysis in disease ecology is the identification of the locations where disease cases are likely to occur and the factors driving these patterns. Meeting this goal requires assessment of the modes of transmission of the disease, as well as the interactions between various hosts, vectors, and the landscape which they inhabit. Most ecologists take an a priori approach, seeking mathematical models of the underlying dynamics that produce patterns mirroring those observed in data (Hudson et al. 2001, Collinge and Ray 2006). In contrast, many public health researchers take an a posteriori approach using statistical techniques to quantify patterns and associations within observational data sets (Halloran 1998). The two epistemological approaches have coexisted at least since the ground-breaking work in both paradigms of Sir Ronald Ross regarding infectious disease in general and malaria in particular in the early 1900's (Ross 1905). However, there has been relatively little cross-fertilization between the two until relatively recently (Hilborn and Mangel 1997, Halloran 1998, Burnham and Anderson 2002).

The growing availability of georeferenced health and environmental data has led to an increased use of geographic information science within disease ecology and often provides a meeting ground between the a priori and a posteriori points of view, further facilitating a more holistic strategy in quantitative methods for disease ecology and related fields such as conservation medicine (Aguirre 2002) and spatial epidemiology (Ostfeld et al. 2005).

In particular, geographic information systems and science provide multiple mechanisms for exploratory spatial analysis in disease ecology. First, basic layering and buffering operations provide insight into the construction of mathematical models by revealing geographic associations between disease incidence, land forms, populations, and local habitats. Second, geographic display of model-based predictions links local observations with model output for assessing fit. Finally, and perhaps most importantly, the geographic display of model output allows statistical a posteriori assessment of model-driven a priori associations by matching spatial patterns arising within the modeled and observed outcomes.

Scientific Fundamentals

The basic components available for spatial analysis in disease ecology involve quantitatively linking the elements of dynamic models of disease transmission and associated measurable data. Beginning with dynamic models of disease transmission, disease ecology typically involves a combination of models of infectious disease transmissions and population dynamics. Halloran (1998) provides a thorough description of what typically are referred to as "SIR" or "SEIR" models where individuals within the population are considered to move between several states: "S" defines those individuals who are not yet infected but are susceptible to infection, "I" defines individuals who are infected (and perhaps, but not always, infectious), and "R" defines individuals who are "recovered" or "removed" through death or immunity. SEIR models incorporate an additional state ("E") for individuals exposed to the pathogen but not yet infected or infectious. One next defines a model, typically based on differential equations, defining the rates of transition from state to state. Such models have seen wide popularity and are used to model a variety of diseases in a variety of settings.

Such models provide theoretical insight into the progression of disease outbreaks and offer testbeds for various intervention scenarios (e. g., does vaccinating children or older adults have a greater impact on the final number of individuals infected during an influenza outbreak?). However, many models are parameterized based on quantities of biological interest (e. g., the basic reproductive number defined as the average number of cases a single infectious individual would infect in a totally susceptible population), or simplifying assumptions (e. g., randomly mixing populations of individuals) that limit connection to data observed in local settings. Halloran (1998) nicely illustrates how the elegant mathematical structure of such models often clashes with the realities of observed data introducing complications in statistical analysis. Moving to a geographical setting raises even more complications such as spatial heterogeneities in population density, interaction rates between "S"s and "I"s, and the influence of local environmental factors.

The statistical analysis of disease ecology data is complicated by the often non-linear models driving disease dynamics and the correlations between disease events induced by the infectious nature of the disease in question. Such correlations arise during the course of an epidemic as the disease passes from individual to individual in space and time, perhaps through intermediate hosts such as vectors or reservoir species.

Recent developments in statistical methodology include the development of hierarchical models allowing the analyst to incorporate uncertainty both in the observations and in the underlying model driving observed dynamics. The general conceptual structure of the hierarchical approach involves three primary stages of the underlying stochastic model: the data model, the process model, and the parameter model. To begin, the data model is a probabilistic description of the data given the underlying process and a set of parameters, i. e., if the underlying dynamic model was known, how would observations arise? Next, it is necessary to define the process model, that is, the set of possible underlying dynamic processes of interest given a set of model parameters (e. g., local transition rates between disease states and the basic reproductive number associated with the disease of interest). Finally, in the Bayesian framework, it is necessary to define prior distributions for each of the data and process parameters. The overall analytic frameworks explores the posterior distribution of these parameters which is proportional to the product of the three hierarchical stages, i. e., [data | process, data parameters]* [process | process parameters]*[data parameters, process parameters] using the bracket notation to denote any general probability distribution and | to denote conditional probabilities. The hierarchical structure not only links process and data models, but also provides mechanisms for incorporating spatial and spatio-temporal correlations within the prior distributions for data and process parameters. Wikle (2003) provides an excellent introductory review of such models and their applications in ecology and climatology.

The hierarchical framework has some conceptual appeal for its linkage of mathematical and statistical modeling, but this appeal can come at a fairly high computational cost. Implementation typically relies on computationally-intensive Markov chain Monte Carlo sampling of the posterior distribution through iterative updates of model parameters. While increasingly common, such approaches are not yet a standard component in statistical or geographic information system software, thereby requiring expertise spanning geographic information science, mathematical modeling, and statistical computing, often best met through interdisciplinary teamwork.

Key Applications

To date, key applications in exploratory spatial analysis of disease ecology data fall primarily into one of the three areas mentioned above. At the risk of oversimplification, it is possible to categorize general trends in disease ecology applications from each of the three classes of analysis. Most geographic information system-based strategies rely on remote sensing data such as precipitation or vegetation cover to define potential habitats for disease vectors or reservoir species, then layer human residences or activity patterns in order to define "risk maps" of areas with high and low propensity for vector-to-human transmission. In contrast, most approaches based on statistical analysis of public health reporting data focus on reported human cases and seek to quantify associations with local factors, some perhaps involving layered remote sensing data. Finally, most spatial ecology approaches rely on stochastic or deterministic models of dynamic spread to predict case locations, then compare these predicted patterns to those observed in the public health data. The analytic stereotypes listed here reveal both the overlap in goals and data, but also highlight the different emphasis placed on different data and modeling pieces by the three different disciplinary approaches.

More recent key efforts begin to lean on more than one framework and brief examples are provided here. Brownstein et al. (2003) expand the risk map format above in an investigation of the spatial distribution of Lyme disease vectors in the United States. Their approach incorporates local climate information based on statistical prediction from fixed sampling stations and links the predictions to observed vector abundance through the use of logistic regression models incorporating spatial correlation. This approach links state-of-the-art geographic information science with sophisticated spatial statistics but requires linking data between a geographic information system and multiple statistical packages in order to provide all of the computation necessary.

A series of papers regarding the spatio-temporal spread of raccoon rabies in the northeastern United States also provides examples of linking geographic information science, mathematical modeling, and spatial statistics. Smith et al. (2002) proposed a cellular automata model of the spread of raccoon rabies across the state of Connecticut. In follow-up work, Waller et al. (2003) use spatial statistics to assess the fit of the mathematical model a posteriori, not only comparing predicted and observed outcomes, but also comparing numerical summaries of the spatial patterns generated by various competing models (e. g., including or not including a delay in spread associated with crossing a river). Finally, Russell et al. (2004) utilized the model of Smith et al. (2002) to provide a priori predictions of the spread of the outbreak through the state of New York.

Future Directions

Spatial analysis in disease ecology provides a meeting ground for individuals from multiple disciplinary backgrounds and, as the illustrations cited above reveal, research in the area increasingly borrows across disciplinary borders. Such interactions provide broader insight and a larger set of analytic tools to address the underlying questions of interest but at the same time require methodological and computational flexibility to make best use of the available methods, models, and data to meet this goal.

Much future research remains in order to provide a comprehensive set of analytic tools providing quantitative links between landscape patterns and disease incidence and prevalence in space and time. Current methods often rely primarily on geographical information systems-based overlays of landscape and health data, mathematical models of dynamics in space and time, or statistical summaries from regression-type models of association. In order to best address the issues raised above, new approaches are required which utilize elements of all three paradigms in order to fully understand the forces and influences driving observed patterns in emerging diseases as well as to suggest and evaluate potential public health intervention strategies.

Cross References

► Data Analysis, Spatial
► Exploratory Visualization
► Geographic Dynamics, Visualization And Modeling
► Hierarchical Spatial Models
► Patterns in Spatio-temporal Data
► Processes and Events
► Statistical Descriptions of Spatial Patterns

Recommended Reading

Aguirre, A.A., Ostfeld, R.S., Tabor, G.M., House, C., Pearl, M.C.: Conservation Medicine: Ecological Health in Practice. Oxford University Press, Oxford (2002)

Bailey, T.C., Gatrell, A.C.: Interactive Spatial Data Analysis. Longman Scientfic and Technical, Essex (1995)

Brownstein, J.S., Holford, T.R., Fish, D.: A climate-based model predicts the spatial distribution of the Lyme disease vector *Ixodes scapularis* in the United States. Envrion. Health Persp. **111**, 1152–1157 (2003)

Burnham, K.P., Anderson, D.R.: Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach, 2nd edn. Springer, New York (2002)

Collinge, S.K., Ray, C. (eds.): Disease Ecology: Community Structure and Pathogen Dynamics. Oxford University Press, Oxford (2006)

Haining, R.: Spatial Data Analysis. Cambridge University Press, New York (2003)

Halloran, M.E.: Concepts in infectious disease epidemiology. In: Rothman, K.J., Greenland, S. (eds.) Modern Epidemiology, 2nd edn. Lippincott-Raven, Philadelphia (1998)

Hilborn, R., Mangel, M.: The Ecological Detective: Confronting Models with Data. Princeton University Press, Princeton (1997)

Hudson, P.J., Rizzoli, A., Grenfell, B.T., Heesterbeek, H., Dobson, A.P. (eds.): The Ecology of Wildlife Diseases. Oxford University Press, Oxford (2001)

Ostfeld, R., Glass, G., Keesing, F.: Spatial epidemiology: an emerging (or re-emerging) discipline. Trends Ecol. Evol. **20**, 328–336 (2005)

Ross, R.: A logical basis of the sanitary policy of mosquito reduction. Science **22**, 689–699 (1905)

Russell, C.A., Smith, D.A., Waller, L.A., Childs, J.F., Real, L.A.: *A priori* prediction of disease invasion dynamics in a novel environment. Proceedings of the Royal Society: Biol. Sci. **271**:1534, 21–25 (2004)

Smith, D., Lucey, B., Waller, L.A., Childs, J.E., Real, L.A.: Predicting the spatial dynamics of rabies epidemics on heterogeneous landscapes. Proceedings of the National Academy of Sciences, USA. **99**, 3668–3672 (2002)

Waller, L.A., Smith, D., Real, L.A., Childs, J.: Monte Carlo assessments of goodness of fit for ecological simulation models. Ecol. Model. **164**, 49–63 (2003)

Wikle, C.: Hierarchical models in environmental science. Int. Stat. Rev. **71**, 181–199 (2003)

Exploratory Visualization

Menno-Jan Kraak
Department of Geo-Information Processing,
ITC – International Institute for Geoinformation
and Earth Observation, Enschede, The Netherlands

Synonyms

Geovisualization; Exploratory cartography; Scientific visualization; Information visualization; Visual data exploration; Visual thinking; Private visual thinking; Reexpression; Temporal ordered space matrix

Definition

Exploratory visualization is the process that involves a discipline expert creating maps and other graphics while dealing with relatively unknown geographic data. These maps are generally for a single purpose, and function as an expedient in the expert's attempt to solve a (geo) problem. While dealing with the data, the expert should be able to rely on cartographic expertise, which allows her or him to look at the data from different perspectives to create a fresh view on the data. As such, the maps and graphics, which should be available in an interactive viewing environment, stimulate the visual thinking of the expert and should support (spatial) decision making.

Historical Background

Visualization means to make visible. Traditionally the cartographic discipline has been active with the visualization of geospatial data. Maps were designed and produced with the presentation of geospatial data in mind. Here, the starting point is facts to be presented, and based on the nature of the data and according the cartographic theory, one would select an appropriate visualization technique. This would result in high-quality visualizations, the single best map. In this process, the emphasis is on map design. However, recent technological developments have given the word visualization an enhanced meaning. The word is linked to the specific ways in which modern computer technology can facilitate the process of "making visible" in real time. The key words here are *interaction* and *dynamics*. This new meaning leads to the term exploratory visualization. The starting point is unknown data without any preconceived ideas about them. The process is an interactive, undirected search for structures and trends. This results in visualizations that, based on different alternative views, can provide a hypothesis. The emphasis is on map use. These developments have been influenced and stimulated by other disciplines. In the 1990s, scientific visualization [1] contributed to the enhanced meaning of visualization to facilitate the process of making data visible in real time in order to strengthen knowledge. The relations between the fields of cartography and geographic information science, on the one hand, and scientific visualization on the other have been discussed in depth in [2,3]. Next to scientific visualization, which deals mainly with medical imaging, process model visualization, and molecular chemistry, another branch of visualization that influenced mapping can be identified: information visualization [4,5]. It focuses on visualization of non-numerical information. Today, both presentation (the communication role of the map) and exploration (the thinking role of the map) are part of the cartographic realm. Communication is described as "public visual communication" since it concerns maps aimed at a wide audience. The cartographic discipline has developed a whole set of design guidelines to realize the most suitable map that offers insight into spatial patterns and relations. Many textbooks and papers have been written on this topic [6,7,8,9]. Thinking is defined as "private visual thinking" because it is often an individual playing with maps and other graphics to study geospatial patterns, relationships, and trends in the data to determine its significance.

Scientific Fundamentals

Environment

One important approach in an exploratory visualization environment is to view geospatial data sets in a number of alternative ways, e. g., using multiple representations without constraints set by traditional techniques or rules. This should avoid the trap described in [10], where it is claimed that "most researchers tend to rely on well-worn procedures and paradigms…" while they should realize that "…creative discoveries, in both art and science, often occur in unusual situations, where one is forced to think unconventionally." This is well described by Keller and Keller [11], who, in their approach to the visualization process suggest removing mental roadblocks and taking some distance from the discipline in order to reduce the effects of traditional constraints. Why not choose an alternative mapping method? For instance, show a video of the landscape next to a topographic map accompanied by a three-dimensional (3D) map, or use a cartogram instead of a choropleth map. New, fresh, creative graphics could be the result. They might also offer different insights and would probably have more impact than traditional mapping methods.
This means a combination of multiple maps and graphics, preferably in an environment that allows access via multiple coordinated views. This would make it possible to

browse the data in a familiar data view and see new patterns in other views, or work in the alternative view using the familiar view as a reference. Several such working environments are known [12,13,14]. One of the important concepts of visual data exploration was introduced in [15] wherein is described the term brushing. Its main characteristic is when the selection of an object in a map view automatically highlights the corresponding elements in the other views. Depending on the view in which the object is selected, there is geographical brushing (clicking in the map), attribute brushing (clicking in the diagram or table), and temporal brushing (clicking on the time line). As such, the user gets an overview of the relations among geographic objects based on location, characteristics and time.

Functionality

To decide about functionality required to support the exploratory visualization process, its objective must be considered. This can be described as finding patterns and relationships in the data that can help answer questions about a geographic phenomenon or process. This process can be split into a set of (sequential) tasks and operations that are needed to meet the goals of the data exploration. The tasks could include checking the spatial positioning of elements of interest in order to verify spatial proximity among different elements, and verifying their spatial density. A comprehensive list includes [16]:

- Identify: to establish the collective characteristics by which an object or cluster of objects is distinctly recognizable (what?).
- Locate: to determine the absolute or relative position (where?).
- Distinguish: to recognize as different or distinct.
- Categorize: to place in specifically defined divisions in a classification; this can be done by color, position, type of object (shape).
- Cluster: to join into groups of the same, similar or related type.
- Distribution: to describe the overall pattern. This is closely related to cluster in the same way that locate and identify are related. The cluster operation asks that the groups be detected, whereas the distribution operation requires a description of the overall clustering.
- Rank: to give an order or position with respect to other objects of like type.
- Compare: to examine so as to notice similarities, differences, or order at different spatial locations.
- Associate: to link or join in a relationship.
- Correlate: to establish a direct connection (correlation).

To be able to execute these tasks, it is necessary to be able, via the interface, to realize the following.

Basic Display Map displays need tools to allow the user to pan, zoom, scale, transform and rotate the image contents. These geometric tools should be available and independent of the dimensionality of the displayed geospatial data. Rotating is a particular need when displaying 3D maps, since it allows views of objects that otherwise might be hidden by other objects. The Google Earth interface is a good example of these functions. Next to the interface options a multitude of generic graphic representation function should be available. Among them the option to switch the map layer on or off, and change the colors and other graphic attributes of the objects represented.

Navigation and Orientation This involves the keys to the map. At any time, the user should be able to know where the view is located and what the symbols mean. This means that a small overview map should be available on demand indicating the location of the current map view. In addition, the coordinates at the cursor should be indicated. A map legend should be available on demand.

Query Data During any phase of the visualization process, the user should have access to the geospatial database behind the graphics to query the data. The questions should not necessarily be limited to simple whats, wheres, or whens. Clicking a geographic object in the map reveals the information available in the database, as well as possible the hyperlinks that are attached to the object.

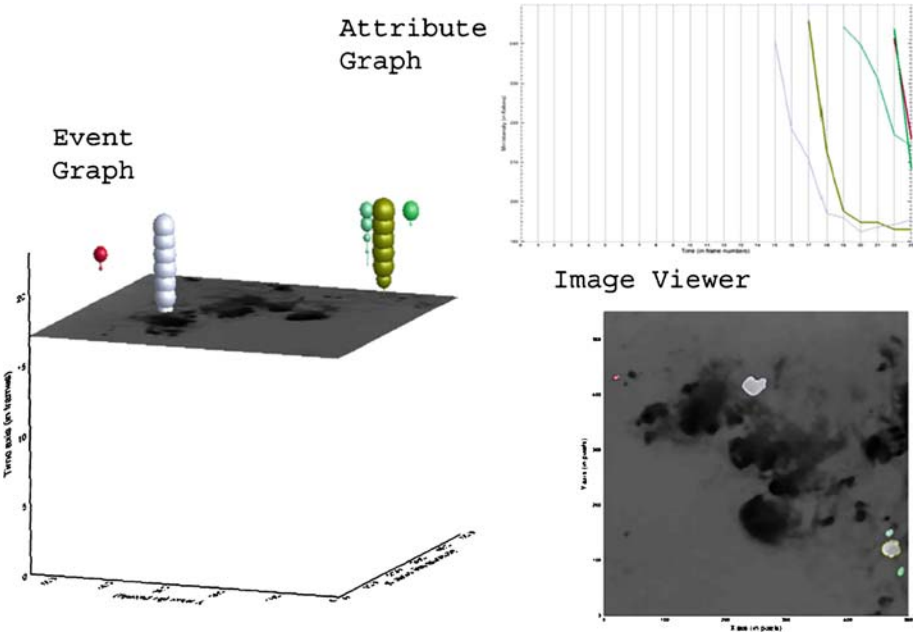
Multiscale Combining different data sets is a common operation in an exploratory environment. The chance that these sets will have the same data density or the same level of abstraction is unlikely. Generalization operators to solve these multiscale problems remain necessary, if only to make sure zoom-in and zoom-out operators result in sensible maps.

Reexpression To stimulate visual thinking, an unorthodox approach to visualization has been recommended. This requires options to manipulate data behind the map or offer different mapping methods for displaying the data. An example of data manipulation is the application of several classification systems, while the use of different advanced map types to view the same data represents display differences. These represent dynamic processes that might use an animation.

Key Applications

Time Series of METEOSAT Images and Convective Cloud Discovery

Meteosat offers an almost constant stream imagery for meteorologists. Working with these time series is impor-



Exploratory Visualization, Figure 1 Time series of METEOSAT images and convective cloud discovery: Functionality of the linked representations: selecting the object trajectory in any of the three views results in highlighting it in others. Selection indicates developing precipitating cloud (clearly visible in the *attribute graph*)

tant for weather prediction. It might for instance help to reveal one particular type of precipitating cloud, namely convective clouds, which are often associated with storms and severe weather conditions. The purpose of search or exploration is to find patterns, trends and relationships in these images in order to detect and predict the behavior of the convective cloud. One way of supporting the exploratory process is by developing appropriate visualization methods. Images are currently explored by animating image sequences with user controlled interactions (play, stop, change rate of animation etc.). However, evidence from the evaluation studies show rather mixed trends. Despite being interactive, animations still lead to information overload, limiting their exploratory usage. To improve this situation, a visual environment consisting of three linked interactive views, each designed to show different characteristic of the tracked object paths, is proposed (see Fig. 1) [17]. It contains event graphs, an (animated) image viewer and an attribute graph. Each of the views is linked to the other via the extracted object paths of the clouds.

This approach should overcome the main factors that limit the exploratory use of animations for the study of convective clouds: data complexity and animation design. It has options to generate more abstract and selective representations. In addition, the visual methods are linked to computational methods, particularly feature tracking. The cloud extracting and tracking can reduce the initial complexity of time-series image data sets: in the prototype each cloud feature can be described in terms of its attributes (position, size, image intensity, etc.) and its lifetime. This offers explicit quantitative, as well as abstract and richer graphical representations, of each cloud's evolution. In particular, the proposed multiple views in combination with the traced cloud paths can show the essence of the object's evolution and history. This means querying the object path in one view results in highlighting the path in the other linked views. Abstract representations help the user to search for objects of interest. For example, convective clouds are revealed by mapping the mean intensity attribute into radius and color of spheres in the event graph (Fig. 1). Convective clouds are also found by the direct manipulation of intensity trajectories in the attribute graph (Fig. 1), and selective interactivity assists the user in focusing the attention on the particular type of clouds. With the proposed functionality, the user can pay more attention to higher order visual tasks: comparison and its related activities, exploration of spatial data and hypothesis generation.

Visual Reasoning with Public Transport Data

Public transport systems (PTS) carry millions of passengers, everyday and everywhere. The movement of people (flow) in PTS is determined by the transport system (network layout, frequency of trips, etc.) and activity centers (such as work, schools and shopping centers).This flow varies in space and time forming varied spatiotemporal patterns. Identifying these patterns will provide insight on existing demand for PTS and its relationship to the activity system of people. The design of a visualization environment that enables an analyst to visualize the PTS data intuitively in a number of alternate ways and explore them interactively is discussed in [18]. The reasoning done by the PTS analysts using the PTS data is considered for designing visual tasks, visual representations and interaction techniques. The prototype of the visualization environment is presented with discussion on how the reasoning is done through it and preliminary usability evaluation of it.

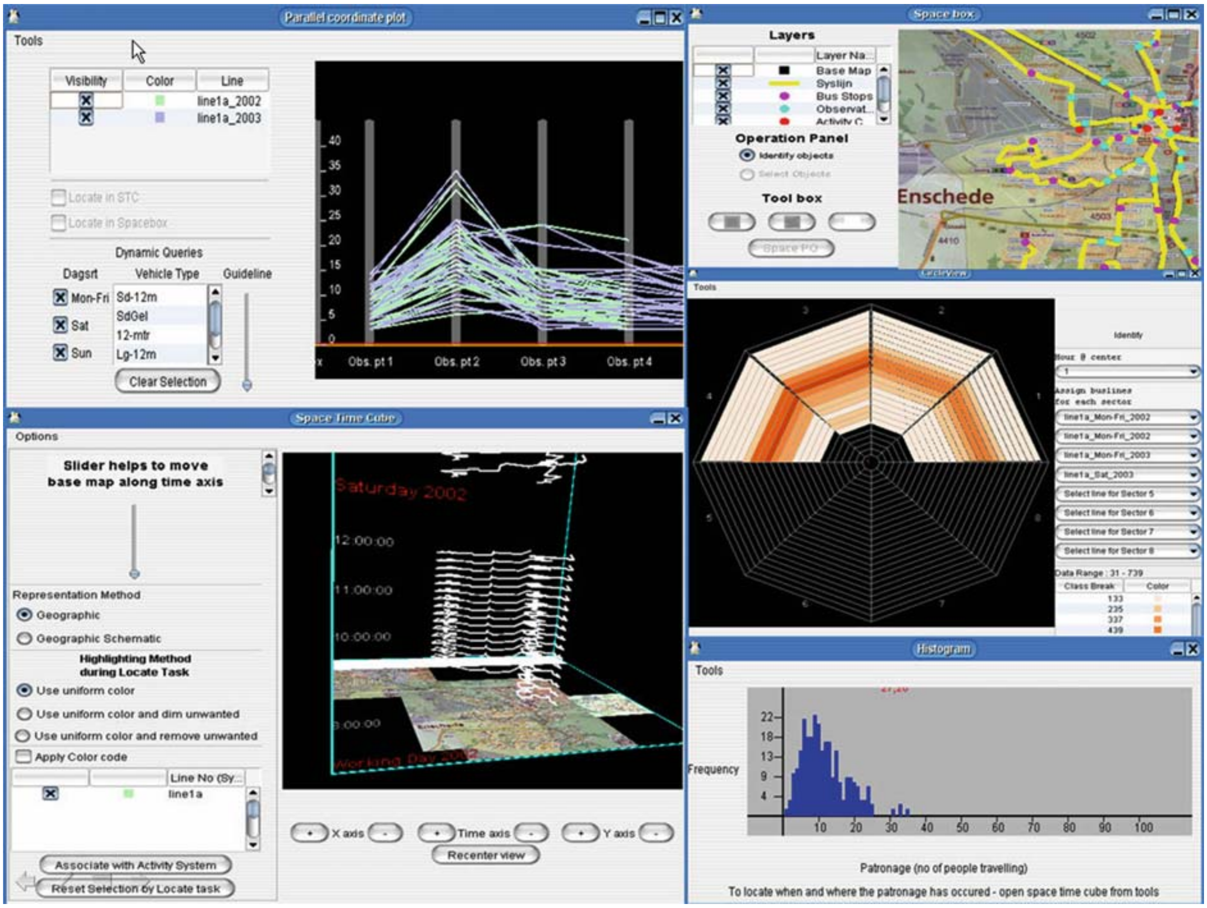
As with the other application, the basic components of geospatial data are guiding the system design. Translating the location, attribute and time to PTS data allows the where (PTS network), when (PTS schedule) and what (PTS flow data) to be distinguished. For this project, PTS data was obtained from the Connexxion bus transport system, one of the major bus service operators in the Netherlands. The data concerned the number of passengers traveling in their network collected through a survey conducted every year in November to assess the PTS performance and make decisions about bus line operations. This data is aggregated to represent PTS flow for working days, Saturdays and Sundays for a year (the bus routes have different schedules for working days, Saturdays and Sundays). Trips are uniquely identified by trip number. The bus line elements trips, path, and number of stops/observation points are different for working day, Saturday and Sunday. In addition, there are changes in bus line element every year. For this work, the data used was for ten bus routes for the years 2002 and 2003.

The analytical reasoning to reveal known and discover unknown spatiotemporal patterns is supported by a set of linked visual representations such as the map, the space–time-cube (STC), histogram, parallel coordinate plot, and circle view, each designed to bring out the attributes of spatial, temporal and spatiotemporal variations in PTS flows (see Fig. 2). Maps are the best visual method to represent geography since they provide an overview and insight into spatial relations and patterns. In geography, the STC is used to represent the movement of people as trajectories in space and time, to understand people's travel behavior. The PTS flow data provides information about number of people traveling in the PTS between consecutive observation points for the PTS trips. Variation in this flow data at the observation point level provides information about how often PTS are overcrowded or underutilized between consecutive observation points of PTS trips. A histogram provides a graphical summary of the distribution of a univariate data set. Apart from showing the distribution,a histogram also helps to identify outliers (exceptional cases in the distribution, if any). Variation in PTS flow data in space provides information about PTS flow pattern "as the PTS make trips." A trip can be represented as a set of observation points, and hence can be considered as multivariate spatial objects. The parallel coordinate plot is often used to represent the multivariate data. A PTS route could have many trips with different sets of observation points for a day. Variations in flow data for a PTS route at different times of a day will help to detect the flow pattern during peak/off-peak hours. The circle view is suggested to visualize the changes in attribute values over the time. Circle View helps to compare and identify patterns, exceptions and similarities in the continuous data, changing their characteristics over time.

Beach Management Application

At the coastal barrier islands between the Wadden Sea and the North Sea in the Netherlands, geomorphological processes such as erosion, transport and sedimentation of sandy materials are causing major changes along the coast of the islands. The problems are related to the location and quantification of beach areas that require nourishment to preserve hinterland to be flooded and have been described in earlier studies. For beach management purposes, the ministry of Public Works divides the beach areas into compartments. Compartments are the regions between two transects. Each compartment has two boundaries to its adjacent compartment, a beach–sea boundary and a beach–dune boundary. To determine which beach compartments are suitable for beach nourishment, a fuzzy classification has been performed based on elevation, vegetation growth, soil wetness and erosion. Next to this classification, spatial data quality elements of interest for the beach management application are defined, i. e., positional accuracy, thematic accuracy, temporal accuracy and data completeness. It is easy to depict a map with beach compartments suitable for nourishment. However, trends and associations between compartments, and changes in time and quality elements involved in the decision making, are more complicated to visualize. Exploratory visualization can assist.

The solution is a prototype of three dynamically linked views, each representing one of the spatial data's components [19]. These are the map view (location), the parallel coordinate plot view (attributes), and the temporal ordered space matrix view (time). The temporal ordered space matrix (TOSM) is a visualization technique like a kind of schematized gridded map whereby the rows in the matrix represent time, the columns represent geographic units, and the individual cell can be color-coded according the value of user-defined attributes. A direct link exists between the geographic units in the animated view and the time view. The geographic units are ordered according to a linear principle. The basic aim of an interactive exploratory visualization is to provide a visual environment that presents data to the user in such a way that it promotes the discovery of (inherent) patterns and relation-

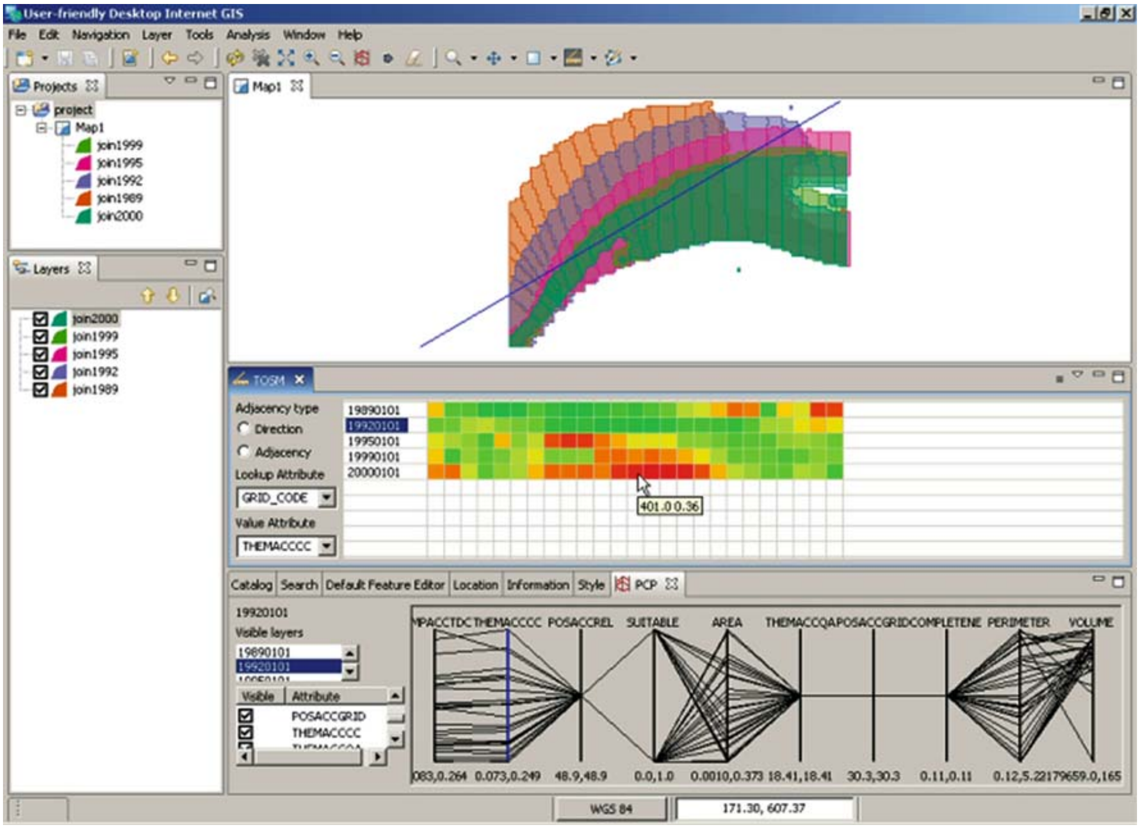


Exploratory Visualization, Figure 2 Visual reasoning with public transport data: example of the different linked views that allow visual exploration. *Clockwise from the upper right*: the map, the circle view, a histogram, the space–time-cube, and a parallel coordinate plot

ships. Extending this functionality for multivariate spatiotemporal applications will lead to insights concerning the dynamic interactive behavior of features (i. e., natural objects or geographical units). Insight will be obtained by identification of changes in features using temporal visualization techniques and by comparison to other variables explaining the cause of these changes by means of multivariate visualization techniques. Hence, the visualization prototype provides multivariate representations, temporal visualization techniques and dynamic interactive functions to allow the user to interact with the data. The TOSM can easily depict beach compartments from southwest to northeast as columns and time of data capture as rows. Each cell reflects an attribute such as nourishment volumes or one of the above-mentioned spatial data quality elements (see Fig. 3). Decision makers in beach management have a distinct overview of trends, relationships and spatiotemporal patterns between compartments and attributes. Trends in nourishment volumes can be examined, as well as their relations towards data quality elements. Hence, visualizing this application in the TOSM is helpful to understand and explore these relationships and trends.

Future Directions

It should now be obvious that maps are no longer the traditional type that most of us know well. Although the traditional role of a map in presenting data is recognized and still valuable, the map should also give us a flexible interface to geospatial data. Maps offer interaction with the data behind the visual representation, and additionally maps are instruments that encourage exploration. As such, maps are used to stimulate (visual) thinking about geospatial patterns, relationships, and trends. The context where maps such as this operate is the world of visual exploration, also known as geovisualization, described in Exploring Geoinformation [20] as a loosely bounded domain that addresses the visual exploration, analysis, synthesis, and presentation of geospatial data by integrating approaches from disciplines including cartography with those from



Exploratory Visualization, Figure 3 Beach management application: screenshot of the prototype with beach management application data. *Top view* depicts the animated map environment, the *middle view* shows the temporal ordered space matrix (*TOSM*), and the *bottom view* illustrates the parallel coordinate plot

scientific visualization, image analysis, information visualization, exploratory data analysis, visual analytics, and geographic information science. In practice, geovisualization activities are viewed in the context of the geoprofessional's working approach. Geodata and software from all types of sources are now linked in geodata infrastructures established on local, national, and international levels. The exchange of geodata and geoservices follows the standards of the Open Geospatial Consortium (see http://www.opengeospatial.org/. Users solve their problems with services and data available via the infrastructure and no longer need to have all data and software locally available. Google Earth is a good example, where the satellite and map data are streamed from external servers, and can be locally combined with our own data for display.

Cross References

► Visualizing Constraint Data
► Web Mapping and Web Cartography

Recommended Reading

1. McCormick, B., DeFanti, T.A., Brown, M.D.: Visualization in Scientific Computing. Comput. Graphics. **21**, (1987)
2. Hearnshaw, H.M., Unwin, D.J., (eds.): Visualization in Geographical Information Systems. Wiley, London (1994)
3. MacEachren, A.M., Taylor, D.R.F., (eds.): Visualization in Modern Cartography. Pergamon, London (1994)
4. Card, S.K., MacKinlay, J.D., Shneiderman, B.: Readings in Information Visualization: Using Vision to Think. Morgan Kaufmann, San Francisco (1999)
5. Spence, R., Information Visualization. Harlow: Addison Wesley/ACM (2001)
6. Kraak, M.J., Ormeling, F.J.: Cartography, Visualization of Geospatial Data, 2nd edn. Prentice Hall, London (2003)
7. Bertin, J.: Semiology Graphique. Mouton, The Hague (1967)
8. MacEachren, A.M.: Some Truth With Maps: A Primer On Design And Symbolization. Association of American Geographers, Washington, DC (1994)
9. Robinson, A.H., Morrison, J.L., Muehrcke, P.C., Kimerling, A.J., Guptill, S.C.: Elements of Cartography. Wiley, New York (1995)
10. Finke, R.A., Ward, T.B., Smith, S.M.: Creative Cognition: Theory, Research, and Applications. The MIT Press, Cambridge, MA (1992)

11. Keller, P.R., Keller, M.M.: Visual Cues, Practical Data Visualization. IEEE, Piscataway (1992)
12. Andrienko, G.L., Andrienko, N.V.: Interactive maps for visual data exploration. Int. J. Geogr. Inf. Sci. **13**, 355–374 (1999)
13. Dykes, J.: Cartographic Visualization: Exploratory Spatial Data Analysis with Local Indicators of Spatial Association Using Tcl/Tk and cdv. Statistician **47**, 485–497 (1998)
14. Haslett, J., Bradley, R., Craig, P., Unwin, A., Wills, G.: Dynamic graphics for exploring spatial data with application to locating global and local anomalies. Am. Statistician **45**, 234–242 (1991)
15. Monmonier, M.: Geographic brushing: enhanching exploratory analysis of the scatterplot matrix. Geogr. Anal. **21**, 81–84 (1989)
16. Koua, E., MacEachren, A.M., Kraak, M.J.: Evaluating the usability of an exploratory geovisualization environment. Int. J. of Geogr. Inf. Sci. **20**, 425–448 (2006)
17. Turdukulov, U.D., Kraak, M.J.: Designing a visual environment for exploration of time series of remote sensing data: in search for convective clouds. Comput. Graphics **31**, 370–379 (2007)
18. Shrinivasan, Y.B., Kraak, M.J.: Visualization of spatio-temporal patterns of public transport system data. In: Proceedings of the 22nd International Cartographic Conference, A Coruna Spain, 9–16 July 2005
19. Kraak, M.J., Vlag, D.v.d.:7 Understanding Spatiotemporal Patterns: Visual Ordering of Space and Time. Cartographica **42**(2), 153–162 (2007)
20. Dykes, J., MacEachren, A.M., Kraak, M.J. (eds.): Exploring Geovisualization. Elsevier, Amsterdam (2005)

Extended Node Tree

► Indexing, X-Tree

Extensibility

► Geography Markup Language (GML)

Extensible Markup Language

MARKUS LUPP
lat/lon GmbH, Bonn, Germany

Synonyms

XML

Definition

Extensible Markup Language (XML) is a simple, very flexible text format derived from **standard generalized markup language** (SGML) (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the web and elsewhere. XML is defined by the World Wide Web Consortium (W3C).

Main Text

XML is a standard for the modeling of data using a tree structure. It consists of rules defining the structure of documents that can store data consistent with the defined structure. XML therefore is a framework for the definition of data structures without describing how an actual document can be processed.

For programs able to process XML, the elements of the documents have to be described in a detailed way. This is especially important for the definition of structural elements and their order inside the document. For this purpose, XML offers two alternatives. The older one is the Document Type Definition (DTD) while XML Schema is the more modern possibility.

In summary, XML is a standard for the definition of arbitrary markup languages that are structurally similar. In the geospatial domain XML is widely used; the most important in this regard is the Geography Markup Language (GML), an XML language for geospatial information.

Cross References

► deegree Free Software
► Geography Markup Language (GML)

References

http://www.w3.org/XML/

Externalization

► Geospatial Semantic Web: Personalisation

Factor Screening Method

► Screening Method

Fastest-Path Computation

DONGHUI ZHANG
College of Computer & Information Science,
Northeastern University, Boston, MA, USA

Synonyms

Fastest route; Driving direction; Shortest paths; Dijktra's shortest path algorithm; A* algorithm

Definition

In the United states, only 9.3% of the households do not have cars. Driving is part of people's daily life. GIS systems like MapQuest and MapPoint are heavily relied on to provide driving directions. However, surprisingly enough, existing systems either ignore the driving speed on road networks or assume that the speed remains constant on the road segments. In both cases, the users' preferred leaving time does not affect the query result. For instance, MapQuest does not ask the users to input the day and time of driving. However, during rush hour, inbound highways to big cities allow much lower speeds than usual. Therefore, a fastest path computed during non-rush hours, which may consist of some inbound highway segments, may not remain the fastest path during rush hour.

Consider a road network modeled as a graph where each node is a road intersection and each edge is a road segment. Let each edge store a speed pattern, e. g., a piecewise-constant function. For instance, in a working day during rush hour (say from 7am to 9am), the speed is 0.3 miles per minute (mpm) and 1mpm at other times of the day.

The ***Time Interval All Fastest Paths (allFP) Query*** is defined as follows. Given a source node s, an end node e, and a leaving time interval at s, the *allFP* query asks to enumerate all fastest paths, each corresponding to a disjoint sub-interval of leaving time. The union of all sub-intervals should cover the entire query time interval.

An allFP query example is: *I may leave for work any time between 7am and 9am; please suggest all fastest paths, e. g., take route A if the leaving time is between 7 and 7:45 and take route B otherwise*.

It is also interesting to solve the allFP problem with an arrival time interval. For instance: *I need to drive from my home to New York International Airport. Please suggest all fastest paths if the anticipated arrival time is between 5pm and 6pm*.

There are two characteristics that distinguish the allFP problem from other shortest/fastest path problems.

- The query is associated with a leaving/arrival time INTERVAL, not a time instant. In fact, if the leaving time were fixed as a time instant, many existing algorithms could be applied, such as the Dijkstra's shortest path computation and the A* algorithm.
- Time is continuous. If time were distinct, e. g., one can only leave precisely at the top of the hour, one could run existing time-instant algorithms multiple times.

Historical Background

Most existing work on path computation has been focused on the shortest-path problem. Several extensions of the Dijkstra algorithm have been proposed, mainly focusing on the maintenance of the priority queue. The A* algorithm [8] finds a path from a given start node to a given end node by employing a heuristic estimate. Each node is ranked by an estimate of the best route that goes through that node. A* visits the nodes in order of this heuristic estimate. A survey on shortest-path computation appeared in [11].

Performance analysis and experimental results regarding the secondary-memory adaptation of shortest path algorithms can be found in [4,13]. The work in [3] contributes on finding the shortest path that satisfies some spatial constraints. A graph index that can be used to prune the search space was proposed in [15].

One promising idea to deal with large-scale networks is to partition a network into fragments. The boundary nodes, which are nodes having direct links to other fragments, construct the nodes of a high-level, smaller graph. This idea of hierarchical path-finding has been explored in the context of computer networks [6] and in the context of transportation systems [5]. In [12], the materialization trade-off in hierarchical shortest path algorithms is examined.

In terms of fastest-path computations, there exists work assuming the discrete model [1,9], the flow speed model [14], or theoretical models beyond road network [10]. The discrete model [1,9] assumes discrete time. The flow speed model work [14] addresses the fastest path query for a given leaving time instant, not a time interval. The theoretical model work [10] suggests operations on travel functions that are necessary without investigating how this operation can be supported.

To compute fastest paths on a road network with a leaving/arrival time interval and with continuous time, one can utilize a novel extension of the A* algorithm. More details are given below.

Scientific Fundamentals

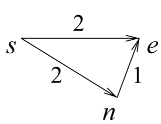
A simple extension to the A* algorithm can NOT be used to solve the allFP query. Let n_0 be the node to be expanded next and let n_0 have three neighbor nodes, n_1, n_2 and n_3. A* picks the neighbor node n_i ($i \in [1..3]$) to continue expanding if the travel time from s to n_i plus the estimated travel time from n_i to e is the smallest. The problem is that since the leaving time is not a fixed time instant, depending on the leaving time instant, different neighbors should be picked.

One possible solution is to expand all such neighbors simultaneously. However, expanding all picked neighbors may result in an exponential number of paths being expanded regardless of the size of the answer set.

Instead, [7] proposed a new algorithm called **IntAllFastestPaths**. The main idea of the algorithm is summarized below:

1. Maintain a priority queue of expanded paths, each of which starts with s. For each path $s \Rightarrow n_i$, maintain $T(l, s \Rightarrow n_i) + T_{est}(n_i \Rightarrow e)$ as a piecewise-linear function of $l \in$ leaving time interval I. Here, $T(l, s \Rightarrow n_i)$ is the travel time from s to n_i, measured as a function of leaving time l. $T_{est}(n_i \Rightarrow e)$ is a lower bound estimation function of the travel time from n_i to the end node e. A straightforward lower-bound estimator is $d_{euc}(n_i, e)/v_{max}$, which is the Euclidean distance between n_i and e, divided by the max speed in the network. A better estimator is called the *boundary node estimator*, which is described later.
2. Similar to the A* Algorithm, in each iteration pick a path from the priority queue to expand. Pick the path whose maintained function's minimum value during I is the minimum among all paths.
3. Maintain a special travel-time function called the *lower border function*. It is the lower border of travel time functions for all identified paths (i. e., paths already picked from the priority queue) that end to e. In other words, for any time instant $l \in I$, the lower border function has a value equal to the minimum value of all travel time functions of identified paths from s to e. This function consists of multiple travel time functions, each corresponding to some path from s to e and some subinterval of I during which this path is the fastest.
4. Stop either when there is no more path left in the priority queue or if the path picked to expand next has a minimum value no less than the maximum value of the lower border function. Report the lower border function as the answer to the allFP query.

Below is a running example. The example involves a simple road network given in Fig. 1. The goal is to find the fastest path from s to e at some time during $I = [6{:}50–7{:}05]$.



$s \to e$: [6-8):1/3
$s \to n$: [6-7):1/3, [7-8):1
$n \to e$: [6-7:08):1/3, [7:08-8):1/10

Fastest-Path Computation, Figure 1 A simple road network. Distances are given on the edges. Speed patterns (#mpm) are given at the right of the network

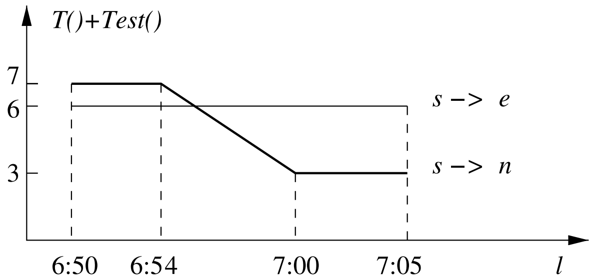
Initially, the priority queue contains only one entry, which corresponds to the unexpanded node s. It has two neighbors e and n. It can be derived that $T(l \in [6{:}50–7{:}05), s \to e) = 6$ min and $T(l \in [6{:}50–7{:}05), s \to n) =$

$$\begin{cases} 6, & l \in [6{:}50–6{:}54) \\ \frac{2}{3}(7{:}00 - l) + 2, & l \in [6{:}54–7{:}00) \\ 2, & l \in [7{:}00–7{:}05] \end{cases}$$

As expressed in step 1 of Algorithm IntAllFastestPaths, in the priority queue, the paths are ordered not by $T()$, but by $T() + T_{est}()$. The functions of the two paths are compared in Fig. 2. Here, $T_{est}(n \Rightarrow e) = 1$ min since $d_{euc}(n, e) = 1$ mile and $v_{max} = 1$ mpm.

According to step 2, the path $s \to n$ is to be expanded next since its minimum value, 3, is smaller than the minimum value, 6, of the path $s \Rightarrow e$.

In general, to expand a path $s \Rightarrow n$, first all the required information for n and its adjacent nodes needs to be retrieved. Then, for each neighbor n_j of n, the following steps need to be followed:



Fastest-Path Computation, Figure 2 Comparison of the functions $T() + T_{est}()$ associated with paths $s \to e$ and $s \to n$

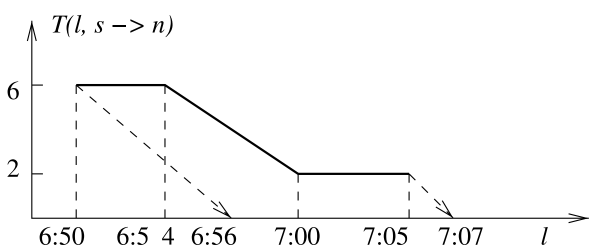
- Given the travel time function for the path $s \Rightarrow n$ and the leaving time interval I from s, determine the time interval during which the travel time function for the road segment $n \to n_j$ is needed.
- Determine the time instants $t_1, t_2, \ldots \in I$ at which the resulting function, i. e., the travel time function for the path $s \Rightarrow n_j$, $T(l \in I, s \Rightarrow n_j)$, changes from one linear function to another.
- For each time interval $[t_1, t_2), \ldots$, determine the corresponding linear function of the resulting function $T(l \in I, s \Rightarrow n_j)$.

In this example, the time interval for $n \to e$ is determined to be [6:56, 7:07], as shown in Fig. 3. At time 6:50 (start of I), the travel time along the path $s \to n$ is 6 minutes. Therefore, the start of the leaving time interval for $n \to e$, i. e., the start of arrival time interval to n, is $6{:}50 + 6\,\text{min} = 6{:}56$. Similarly, the end of the leaving time interval is $7{:}05 + 2\,\text{min} = 7{:}07$.

During the time interval [6:56–7:07], the travel time on $n \to e$, $T(l \in [6{:}56 - 7{:}07], n \to e)$ is

$$\begin{cases} 3, & \text{if } l \in [6{:}56\text{–}7{:}05) \\ 10 - \frac{7}{3}(7{:}08 - l), & \text{if } l \in [7{:}05\text{–}7{:}07] \end{cases}$$

There are two cases that trigger the resulting travel time function $T(l, s \Rightarrow n \to e)$ to change from one linear function to another. In the first simple case, the function $T(l, s \Rightarrow n)$ changes. The time instants at which the resulting function changes are the ones at which $T(l, s \Rightarrow n)$ changes.



Fastest-Path Computation, Figure 3 The time interval, [6:56–7:07], during which the speed on $n \to e$ is needed

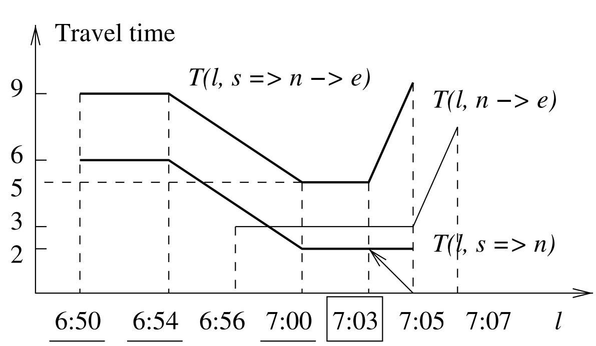
In Fig. 3, these correspond to time instants 6:50, 6:54 and 7:00. In the second trickier case, the changes of the resulting function are triggered by the changes of $T(l, n \to e)$, e. g., at time 7:05. In this example, one can determine that at time 7:03, $T(l, s \Rightarrow n \to e)$ changes. The reason for this is that if one leaves s at 7:03, since the travel time on $s \Rightarrow n$ is 2 minutes, one will arrive at n at 7:05. At that time the travel time function of $n \to e$ changes. To find the time instant 7:03, compute the intersection of the function $T(l, s \Rightarrow n)$ with a 135° line passing through the point (7:05, 0). The time instant 7:03 is the leaving time corresponding to that intersection point.

Now that all the four time instants 6:50, 6:54, 7:00, and 7:03 have been determined, the 4-piece function $T(l \in I, s \Rightarrow n \to e)$ can be derived by combining $T(l, s \Rightarrow n)$ and $T(n \to e)$.

For each l, $T(l, s \Rightarrow n \to e)$ is equal to $T(l, s \Rightarrow n)$ plus $T(l', n \to e)$, where l' is the time at which node n is reached. That is, $l' = l + T(l, s \Rightarrow n)$. The following algorithm should be used to expand a path for every identified time instant $t \in \{t_1, t_2, \ldots\}$ (e. g. 6:50):

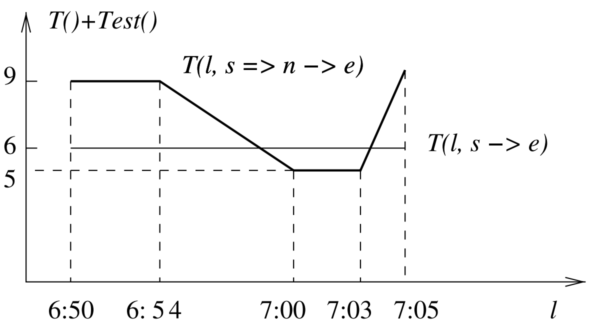
- Retrieve the linear function of $T(l, s \Rightarrow n)$ at time t. Let it be $\alpha * l + \beta$.
- Retrieve the linear function of $T(l', n \to e)$ at time $t' = t + (\alpha * t + \beta)$. Let it be $\gamma * l' + \delta$.
- Compute a new linear function $(\alpha * l + \beta) + (\gamma * (l + \alpha * l + \beta) + \delta)$ which can be re-written as $(\alpha * \gamma + \alpha + \gamma) * l + (\beta * \gamma + \beta + \delta)$. This is the linear function as part of $T(l, s \Rightarrow n \to e)$ for the time interval from t to the next identified time instant.

For instance, the combined function $T(l \in \mathrm{I}, n \Rightarrow e)$, which is shown in Fig. 4, is computed as follows. At $t = 6{:}50$, the first linear function is a constant function 6. Hence, $t' = t + 6 = 6{:}56$. The second linear function starting with 6:56 is another constant function 3. Therefore, the combined function is 9, which is valid until the next identified time instant.

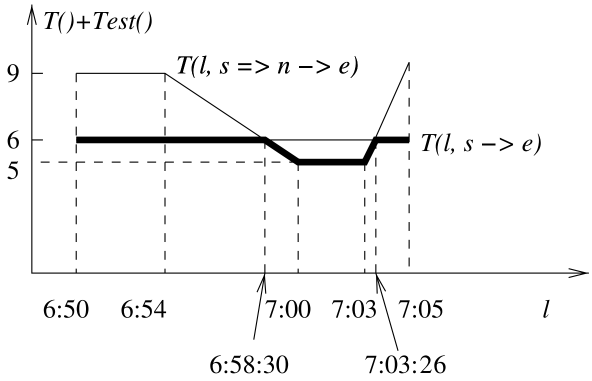


Fastest-Path Computation, Figure 4 The time instants at which $T(l, s \Rightarrow n \to e)$ changes to another linear function and the $T(l, s \Rightarrow n \to e)$ function

At $t = 6{:}54$, the first linear function is $\frac{2}{3}(\mathtt{7:00} - l) + 2$. Therefore, $t' = 6{:}54 + 6 = 7{:}00$. The second linear function is 3. The combined function is $\frac{2}{3}(\mathtt{7:00} - l) + 5$.
At $t = 7{:}00$, the first function is constant 2. At $t' = 7{:}00 + 2 = 7{:}02$, the second function is 3. Thus, the combined function is 5.
Finally, at $t = 7{:}03$, the first function is 2, and at $t' = 7{:}03 + 2 = 7{:}05$, the second function at $10 - \frac{7}{3}(\mathtt{7:08} - l')$. Therefore, the combined function is $2 + (10 - \frac{7}{3}(\mathtt{7:08} - (l + 2))) = 12 - \frac{7}{3}(\mathtt{7:06} - l)$.
After the expansion, the priority queue contains two functions, as shown in Fig. 5. Note that in both functions, the lower bound estimation part is 0 since both paths already end with e.
The next step of Algorithm IntAllFastestPaths is to pick the path $s \Rightarrow n \rightarrow e$, as its minimum value (5 min) is globally the smallest in the queue. An important question that arises here is when to stop expanding, as expanding all paths to the end node is prohibitively expensive. The algorithm terminates when the next path has a minimum value no less than the maximum value of the maintained *lower border function*.



Fastest-Path Computation, Figure 5 The two functions in the priority queue



Fastest-Path Computation, Figure 6 The lower border and the result for Query 3

When there is only one identified path that ends with e, the lower border function is the function of this path. In Fig. 5, $T(l, s \Rightarrow n \rightarrow e)$ is the lower border function. As each new path ending with e is identified, its function is combined with the previous lower border function. For example, in Fig. 6, the new lower border function after the function $T(l, s \rightarrow e)$ is removed from the priority queue is shown as the thick polyline.
The algorithm can terminate if the next path to be expanded has a minimum value no less than the maximum value of the lower border function (in this case, 6). Since the maximum value of the lower border keeps decreasing while the minimum travel time of paths in the priority queue keeps increasing, the algorithm IntAllFastestPaths is expected to terminate very fast. In this example, the set of all fastest paths from s to e when $l \in$ [6:50–7:05] is:

$$\begin{cases} s \rightarrow e, & \texttt{if} \quad l \in \texttt{[6:50-6:58:30)} \\ s \rightarrow n \rightarrow e, & \texttt{if} \quad l \in \texttt{[6:58:30-7:03:26)} \\ s \rightarrow e, & \texttt{if} \quad l \in \texttt{[7:03:26-7:05]} \end{cases}$$

Finally, the boundary-node estimator, which is a lower-bound travel time from n_i to e and is used to improve the efficiency of the algorithm, is described below.

- Partition the space into non-overlapping cells [2]. A **boundary node** [6] of a cell is a node directly connected with some other node in a different cell. That is, any path linking a node in a cell C_1 with some node in a different cell C_2 must go through at least two boundary nodes, one in C_1 and one in C_2.
- For each pair of cells, (C_1, C_2), pre-compute the fastest travel time (function) from each boundary node in C_1 to each boundary node in C_2.
- For each node inside a cell, pre-compute the fastest travel time from and to each boundary node.
- At query time, n_i and e are given. Since any path from n_i to e must go through some boundary node in the cell of n_i and through some boundary node in the cell of e, a lower-bound estimator can be derived as the summation of three parts: (1) the fastest time from n_i to its nearest boundary node, (2) the fastest time from some boundary node in n_i's cell to some boundary node in e's cell, and (3) the fastest time from e's nearest boundary node to e.

Key Applications

The key application of fastest-path computation is road navigation systems. Some examples include Mapquest.com, Local.live.com, and MapPoint. Such systems can produce better driving directions if integrated with traffic patterns and fastest-path computation techniques.

Future Directions

To speed up the calculation, the road network should be partitioned. At the index level, each partition is treated as a single network node and the details within each partition are omitted. Pre-computation is performed to calculate the travel time from each input edge to each output edge. The partitioning can be performed hierarchically. Another direction is that, to simplify the computed fastest paths, the algorithm should be extended to allow the users to provide a maximum number of changes in road names.

Cross References

► Dynamic Travel Time Maps
► Routing Vehicles, Algorithms
► Trip Planning Queries in Road Network Databases

Recommended Reading

1. Chabini, I.: Discrete Dynamic Shortest Path Problems in Transportation Applications. Trans Res Rec **1645**, 170–175 (1998)
2. Chakka, V.P., Everspaugh, A., Patel, J.M.: Indexing Large Trajectory Data Sets With SETI. In: Biennial Conf. on Innovative Data Systems Research (CIDR), Asilomar, CA, USA 2003
3. Huang, Y., Jing, N., Rundensteiner, E.: Spatial Joins Using R-trees: Breadth-First Traversal with Global Optimizations. In: VLDB, Athens, Greece, pp. 396–405 (1997)
4. Jiang, B.: I/O-Efficiency of Shortest Path Algorithms: An Analysis. In: ICDE, pp. 12–19, Tempe, AZ, USA, 1992. IEEE Computer Society (1992)
5. Jing, N., Huang, Y.W., Rundensteiner, E.A.: Hierarchical Encoded Path Views for Path Query Processing: An Optimal Model and Its Performance Evaluation. TKDE **10**(3):409–432 (1998)
6. Kamoun, F., Kleinrock, L.: Hierarchical Routing for Large Networks: Performance Evaluation and Optimization. Comput Netw **1**:155–174 (1977)
7. Kanoulas, E., Du, Y., Xia, T., Zhang, D.: Finding Fastest Paths on A Road Network with Speed Patterns. ICDE, Atlanta, GA, USA, 2006. IEEE Computer Society (2006)
8. Kung, R.M., Hanson, E.N., Ioannidis, Y.E., Sellis, T.K., Shapiro, L.D., Stonebraker, M.: Heuristic Search in Data Base Systems. In: Expert Database Systems Workshop (EDS), pp. 537–548 (1984)
9. Nachtigall, K.: Time depending shortest-path problems with applications to railway networks. Eur J Oper Res **83**:154–166 (1995)
10. Orda, A., Rom, R.: Minimum Weight Paths in Time-Dependent Networks. In: Networks: An International Journal **21** (1991)
11. Pallottino, S., Scutellà, M.G.: Shortest Path Algorithms in Transportation Models: Classical and Innovative Aspects. In: Marcotte, P., Nguyen, S. (ed.) Equilibrium and Advanced Transportation Modelling, pp. 245–281. Kluwer Academic Publishers, New York (1998)
12. Shekhar, S., Fetterer, A., Goyal, B.: Materialization Trade-Offs in Hierarchical Shortest Path Algorithms. In: SSTD, pp. 94–111. Springer, Berlin (1997)
13. Shekhar, S., Kohli, A., Coyle, M.: Path Computation Algorithms for Advanced Traveller Information System (ATIS). In: ICDE, Vienna, Austria, 1993. IEEE Computer Society, pp. 31–39 (1993)
14. Sung, K., Bell, M.G.H., Seong, M., Park, S.: Shortest paths in a network with time-dependent flow speeds. Eur J Oper Res **121**(1), 32–39 (2000)
15. Zhao, J.L., Zaki, A.: Spatial Data Traversal in Road Map Databases: A Graph Indexing Approach. In: Proc. of Int. Conf. on Information and Knowledge Management (CIKM), Gaithersburg, MD, USA, 1994. ACM, pp. 355–362 (1994)

Fastest Route

► Fastest-Path Computation

F

FCC 94-102

► Indoor Localization

Feature Catalogue

Jean Brodeur[1], Thierry Badard[2]
[1] Center for Topographic Information, Natural Resources Canada, Sherbrooke, QC, Canada
[2] Center for Research in Geomatics (CRG), Department of Geomatic Science, Université Laval, Quebec, Canada

Synonyms

Content metadata; Abstract representation of geographic data; Machine readable geographic data

Definition

In the context of geographic information and ISO/TC 211 vocabulary, a feature catalogue refers to a description of an abstraction of reality that may be used to depict one or more geographic datasets. Consequently, it constitutes a classification of phenomena [3] and may be used along with an application schema.

Main Text

A feature catalogue consists of a collection of metadata that provides the semantics and the structure of the objects stored in a geographic database. A feature catalogue includes (1) the names and definitions of feature types, (2) their properties' name and definition including feature attributes, geometry (shapes and specifications, datum, map projection, etc.), temporality (dimensions and specifications, datum, units, resolutions, etc.), operations, and roles, (3) descriptions of attribute values and domains, relationships, constraints, and so on. A feature catalogue may be presented in various forms such a text document, a database, a spreadsheet, etc. Typically, a feature catalogue is available in electronic form to support interoperability of geographic information. Although a feature catalogue addresses the same content of an application schema, they are both complementary in the manner they represent it. Levels of details regarding catalogues (data dictionaries) and schemata (models) are described in the cross-references.

Cross References

► Modeling with ISO 191xx Standards
► Modeling with Pictogrammic Languages

Recommended Reading

1. Brodeur, J., Bédard, Y., Proulx, M.J.: Modelling Geospatial Application Databases using UML-based Repositories Aligned with International Standards in Geomatics. 8th ACM Symposium on Advances in Geographic Information Systems (ACMGIS) (2000)
2. ISO/TC211 ISO19109:2005 Geographic information – Rules for application schema. ISO (2005)
3. ISO/TC211 ISO19110:2005 Geographic Information – Methodology for feature cataloguing. ISO (2005)

Feature Extraction

► Image Mining, Spatial

Feature Extraction, Abstract

Ashok Samal, Sharad Seth
Department of Computer Science and Engineering,
University of Nebraska-Lincoln, Lincoln, NE, USA

Synonyms

Automated map generalization; High-level features; Functional description; Abstract features

Definition

Extraction of features from remotely sensed imagery is an important problem in many application domains. Current technologies for feature extraction for urban scenes rely on physical attributes as the basis for classification. Roads, railroads, rivers, buildings, center pivots, lakes, reservoirs, airports, and canals are typical examples of such features. Although this approach yields important information from imagery, low-level techniques to information extraction, classification and management limit both the analysis model and the quality and quantity of information extracted from the image. They do not yield to a higher-level analysis, which allows for a true understanding of the role that individual and groups of features play in processes akin to human cognition. The reason for their failure is the gap between the representation used for features and the models used for cognitive analyses. The current models use individual physical entities as the basic unit for analysis, but the processes that shape significant events on the earth's surface are affected by many competing and coordinating entities that form the basic building blocks for analysis. These entities, therefore, can be viewed as abstract features and must be used to obtain a deeper understanding of geographic events in both space and time. A high-level feature is defined by the characteristic distribution of its constituent features and also by its relationship with other high and low level features – its geographic context. Figure 1 shows an example of such abstract features.

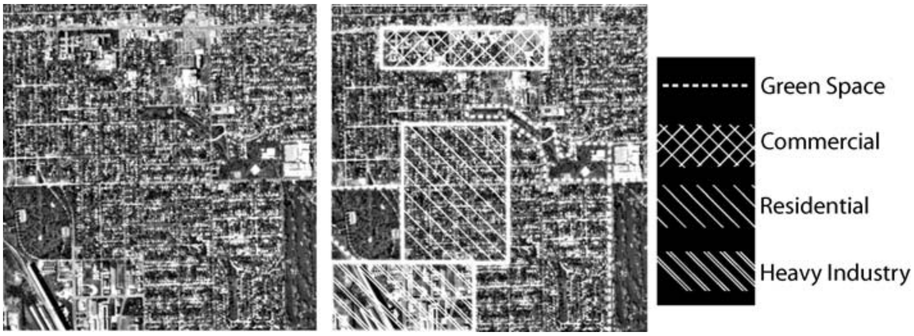
There is a need to extend the classification model into one that more closely parallels the human cognitive system. By organizing features at a higher level than simple physical location, shape and size at a single point in time and space, one can derive a functional description of the geographic space.

Historical Background

Satellite imagery and aerial photograph interpretation has been widely used over the past three decades. It was initially used in military applications for reconnaissance purposes. With the increasing availability and the decreasing cost of such sources of data, it has migrated to many civilian applications, e. g., generations of digital maps, map planning, and surveys of land use and land cover. These remotely sensed images, ranging from the 1-m high-resolution panchromatic images provided by IKONOS and QuickBird to 2.5-m medium-resolution panchromatic images provided by SPOT 5 and the 15-m low-resolution panchromatic images provided by Landsat 7 have become one of the most pervasive sources of spatial data used in the GIS community.

Before their incorporation in GIS applications, images should be delineated and classified to predefined patterns and objects that can be used to build GIS topology useful for analysis, modeling, map making or 3D terrain visualizations. Often, interpretation is manually performed by experienced photo interpreters. Due to the inherent inefficiency in manual interpretation and the limited availability of qualified operators, many images remain unprocessed. This problem will only exacerbate as more high-resolution satellites are launched in the coming years and as other remotely sensed images become available and affordable.

In order to improve the efficiency of interpretation, many fully or partially automated processes have been developed



Feature Extraction, Abstract, Figure 1 USGS Digital Orthophoto Quarterquadrangle (1:12000) Lincoln, Nebraska and some possible abstract features

with their focus dependent on image resolution. For high-resolution images of sub-meter resolution, the emphasis is on extracting individual buildings or the road network in a standard two-step process [2,5,6,8,9,10].

1. Line segments deemed as parts of roads or buildings are extracted by tracking the extreme value of local gradient or the energy of the image intensity.
2. Extracted segments are conceptually grouped according to a road or building model.

For low-resolution images, usually greater than ten meters, the focus has been on methods to extract the boundary of regions that have specific spectral properties or texture patterns. The image segmentation for land cover and land use could be divided into edge-based and region growing approaches. The most common application of such analysis is in drawing the land use or land cover thematic map. There is relatively little research with focus on an integrated approach that seamlessly bridges the gap between the processing of high and low resolution images. In other words, there is no framework to interpret the internal structure of a land cover. For example, in an urban area, one not only would want to find the coverage in terms of residential or commercial areas, but may further want to know whether the covered area is a single family residence or a multi-family residence, or a general office. Better yet, one may want to find the geographical composition of a region, i. e., by building the spatial relation of the blocks within the region. Such a representation of the image has many useful applications including understanding the urban sprawl process, planning the utility locations, managing traffic, and evaluating the loss due to a hazard.

One trend in GIS is to develop an ontology-driven GIS [3], which means the elements of the GIS are not the usual layers of points, lines and polygons, but objects that denote location information, spatial relations, associated events and interfaces between objects. Ontology-driven GIS will facilitate recording or simulating the evolution of an area in a period of time. This proposal for image interpretation fits ithis scheme by providing the spatial description of the ontology.

The focus of the paper is limited to the urban landscape for two reasons. First, urban areas include a large variety of regions occurring in complex configurations that are challenging to interpret. Second, being the result of human activities, urban areas have a certain degree of regularity in the arrangement of spatial objects that can be helpful in geographic model building and automated feature interpretation.

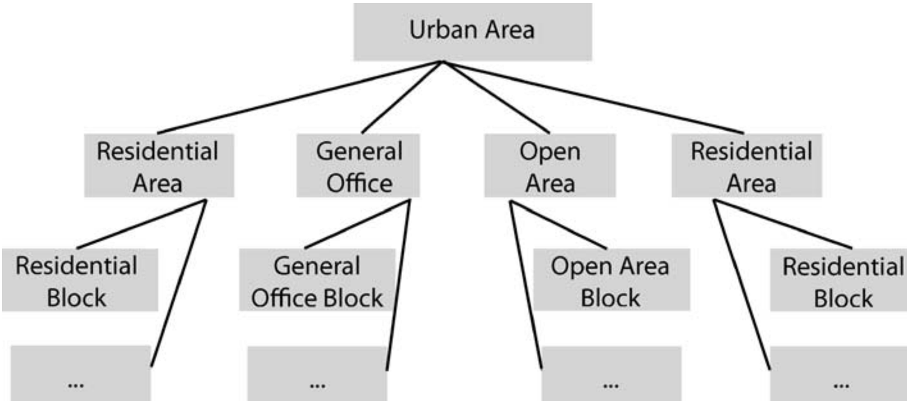
Further, because extensive work on the extraction of roads and buildings already exists, the assumption is that this level of image interpretation is available in vectorized form. The focus is, instead, on analyzing the structure of an urban area.

Human interpreters deduce land use by means of various clues, such as color, texture, shape, size, and the spatial relationship between adjacent objects visible in the image. All these clues of a local region will be denoted as its *context*. While context has been used in traditional interpretation methods to recognize individual buildings, presently it is used for a higher level of image interpretation, e. g., to find a cluster of buildings that have a specific function.

Scientific Fundamentals

Abstract features are rich, diverse, spatially disjoint, temporally correlated and inherently fuzzy. These properties impart tremendous power and pose great challenges to abstract feature extraction. The properties of abstract features are briefly described below.

- The difference between physical and abstract features is not just a matter of scale. In general, abstract features will be defined at broader scales in both time and space. For example, an abstract feature may contain disjoint geographic areas that may display particular characteristics.
- Abstract features are domain dependent. Since the abstract features are tied to concepts, they are defined in the context of an application domain.
- The constituents of an abstract feature may not be spatially contiguous. The power of abstract features is



Feature Extraction, Abstract, Figure 2 The hierarchical composition tree of an urban area

derived from the ability to connect spatially disjoint, but conceptually similar physical features.

- Abstract features are dependent on the temporal dimension. Time is a critical factor in the definition and hence, abstract features can not be obtained by static analysis.
- Abstract features are fuzzy. Processes affecting events on the surface of the Earth can not be characterized with parameters that are certain or hard. Thus, a faithful representation of abstract features must incorporate some uncertainty.

Attributes of abstract features are described in a variety of sources and conflation is an integral part of their extraction. In addition to remotely sensed images, a variety of ancillary data may be used to increase the efficiency and accuracy of the feature extraction process. By drawing upon multiple sources, a more complete description of features may be generated.

The focus of this article is extraction of abstract features for urban areas. The basic element in this analysis is the block, defined as the smallest region that cannot be divided by any road segments. A block includes both its surrounding roads and the buildings inside. In this approach, the category of a block is first determined by checking its global context. If it is a mixed block, it is broken into sub blocks based on the consistency of the global context. Conversely, if adjacent blocks have been determined to be of the same category, they are merged. At the end of the interpretation process, a hierarchical composition tree of an urban area similar to the one shown in Fig. 2 is derived.

Interpretation of the image of the urban area is done in three steps in this approach:

1. Define the classification system for the land cover and land use in the urban area
2. Determine the global context for each category in the classification system
3. Design an algorithm for recognizing each category.

The classification system in remote sensing is organized in a 4-level hierarchy [1]. Levels I and II are generally of interest to users who desire data on a nationwide, interstate or statewide basis. Levels III and IV are used by those concerned about information at the regional, county, or municipal level. USGS has standardized the classification of Levels I and II [1].

Levels III and IV classifications consider the remotely sensed image in more detail and require a substantial amount of supplemental information in addition to the image data. This is one reason for the lack of standardization at these levels. Some examples of Levels III and IV classifications used by local government have been described in the literature [4]. However, these classifications emphasize more on functions of the region that are not available from the remotely sensed imagery. Since the focus here is to only extract the information from the imagery, the classification system has been revised by merging classes that have the same visual appearance but have different functions. For example, commercial office building and public service building categories have been combined into a general office building category. It should be noted that use of ancillary data can only improve this analysis.

To transfer the function directed classification system to an appearance directed system, the boundary of Level II classification is broken and the Level III classification is reorganized. For example, Class 1, described below, corresponds to Category 14 in Level II, Classes 2, 3, and 4 belong to Category 11, and Class 5 matches with Category 12, but Class 9 is a combination of some subclasses of Categories 13 and 15. The classes in the scheme are described below.

1. *Aviation Area (AA)*: The area which is used or is intended to be used, primarily for the take-off and landing of aircraft and any appurtenant areas which are used, or intended to be used, for airport buildings or facilities, including open spaces, taxiways and tie-down areas, hangers, and other accessory buildings. The indication of Aviation Area is the run-

way, and the vast area of grassland around the runway.

2. *Single Family Residence (SFR)*: The area which is composed of lots where each one has one building designed for and contains not more than two separate units with facilities for living, sleeping, cooking, and eating therein. The buildings in this category are of equal size and have a regular layout.
3. *Multi-Family Residence (MFR)*: The area that is a cluster of dwelling units that supports multi-families in a residential setting. Buildings in this area are larger than those in *SFR* and are surrounded by a sufficient parking place. *Multi-Family Residenc*es are always close to *Single Family Residence*s.
4. *Mobile Home Residence (MHR)*: The area that is a cluster of transportable structures built on a chassis and is designed, when connected to the required utilities, to be used as a year-round dwelling unit with or without a permanent foundation. Buildings in *Mobile Home Residence* are smaller than those in *SFR* and the layout is more compact.
5. *Commercial Business District (CBD)*: The area that is the concentration of financial and professional services and major retail outlets, the focus of transport lines, where land use is the most dense and land values are at their highest. Most of this area is covered by man-made objects and it always appears in the center of the city.
6. *Shopping Complex (SC)*: The area used for mercantile establishment consisting of a carefully landscaped complex of shops representing leading merchandisers. *Shopping Complex* is an agglomeration of several large buildings surrounded by a large parking lot.
7. *Common Commercial Service and Office (CCSO)*: The area used for commercial or industrial activities that typically have operating characteristics or traffic service requirements generally incompatible with residential environments, such as equipment sales, custom manufacturing, vehicle storage, or construction services. Buildings in this category are not as big as those in the *Shopping Complex* category, but larger than those in residential and spread along the main streets. The layout of entities in this category is not as compact as those in *Commercial Business District*.
8. *Open Area (OA)*: The area that has very few man-made objects mostly consists of bare land, land covered by vegetation, or bodies of water.
9. *Industrial District (ID)*: The area that is intended for commercial services, research and development, administrative facilities, warehousing and distributions, and manufacturing uses. Buildings in the *Industrial District* are big, but have little space for parking, and this category always appears at the outskirt of the city.

In the definition of each class, some of the spatial characteristics have been described. However, these descriptions of geometric, spatial and other properties should be formalized so that they can be used in an automated system. These properties for a specific scene are called its context or sometimes its signature. The context defines the rules for combining adjacent buildings into a community unit with a specific social function. Context is used as the basis for recognition.

The focus here is on the *Single Family Residence* (*SFR*). Figure 3 presents several examples of the *SFR*, from which many important properties can be extracted.

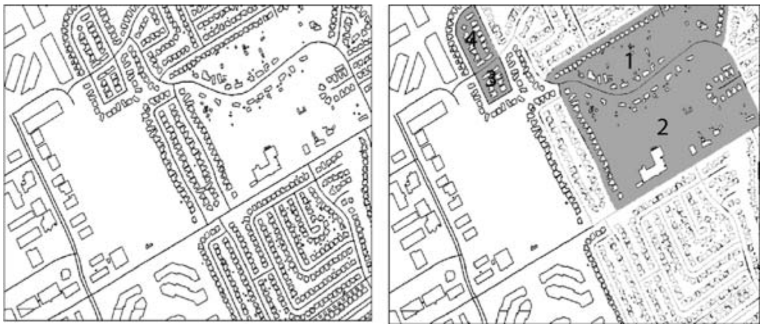
By analyzing the structure of the blocks in the *SFR*, several constraints that can be used as a global context or signature for the *SFR* are derived. Two kinds of constraints are considered: *Scalar* Constraints and *Spatial* Constraints. Scalar constraints describe properties without geometric information like dimension and size. Spatial constraints characterize direction and location relationships between the adjacent objects and their layout of the full block. Examples of each type of constraint follow.

1. *Spatial Constraints*
   a. The *SFR* buildings are approximately at the same distance to their nearest road segments.
   b. The buildings typically are linearly arranged and the distances between buildings are approximately the same.
   c. One of the dimensions of a building is aligned with the nearest road segment.
2. *Scalar Constraints*
   a. The area and dimensions of buildings in a *SFR* block are similar.
   b. The residential buildings in a *SFR* block typically cover only a small part of the total area of the block.

The following algorithm describes the approach to determining if a block is a *SFR* block.

1. Calculate the area of all the buildings in the block.
2. If *(buildings area)/(block area)* > *th1* (a predefined but adaptable threshold), then the block is rejected as an *SFR* and exit.
3. Enumerate all the buildings whose area falls in a predefined range [*t1*, *t2*]. (*t1* and *t2* are derived from statistical analysis of given *SFR* examples).
4. Find the bounding rectangles for all buildings that may be single family housing units. Link the center of these bounding rectangles by following along the road boundary.
5. Find all single family building chains by analyzing the distance to the nearby road network, the distance to adjacent buildings and the direction of the build-

Feature Extraction, Abstract, Figure 3 Example composed primarily of a Single Family Residential Area (SFR)



Feature Extraction, Abstract, Figure 4 An urban scene (*left*) and its interpretation using only scalar constraints (*right*)

ings. A building that satisfies three spatial constraints declared in the previous paragraph will be seen as a candidate for a single family house unit. Start a chain by choosing a candidate randomly and track along the boundary road. If the next building is a candidate, the chain continues, otherwise the chain stops and a new chain starts when the next candidate is found in the track.

6. Project the centers of the two end buildings to the boundary and the boundary between these two projected points is defined as SF-Chain boundary. Calculate the length of the SF-Chain boundary.
7. If Σ (*length of SF-Chain boundary*)/(*length of the whole boundary*) > *th2* (another predetermined threshold), then this block is classified as an *SFR*.
8. Otherwise, this block is classified as a MIXED block.
9. Merge all adjacent *SFR* blocks.

Another approach for building generalization for urban scenes using morphology and Gestalt theory has been proposed by Li, et al. [7].

A Case Study

To show the feasibility and the accuracy of the context directed interpretation, it has been tested with the GIS data of Austin, TX. The population of this city is approximately 660,000 and it covers about 260 square miles. It has 277,000 housing units and is the 16th largest city in the US.

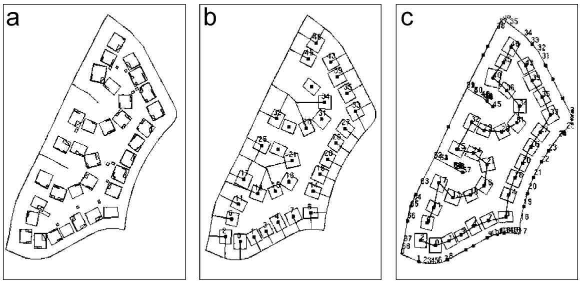
Only some sections of the city are used for testing and reported here. Figure 4 (left) shows part of the dataset used for testing. The long linear structures represent the road network and the small polygons are the footprints of the buildings. The dataset consists of a *Single Family Residence* adjacent to a *General Office* or a *Multi-Family Residence*. Figure 4 (right) shows the results of the interpretation using only two scalar constraints:

1. The building coverage of the block is between 0.15 and 0.35
2. The average size of the building is between 1200 and 3000 square units.

The thresholds are determined manually by studying the properties of several blocks. They can also be extracted by unsupervised methods.

Of all the 33 blocks in Fig. 4, only four SFR blocks (1, 2, 3 and 4) are misclassified. The shaded regions represent the correctly recognized Single Family Residence blocks. The reason for misclassification of Blocks 1 and 2 is that the building density is too low and the misclassification of Block 3 and 4 is due to the high building density. This is a disadvantage of using scalar constraints, because universal thresholds are difficult to accurately determine.

Figure 5 shows several steps during the interpretation using both scalar and spatial constraints. Figure 5a shows the result of finding the bounding box of the building footprints. Figure 5b shows the principal direction of each building; Buildings less than 1200 square units are filtered out. Figure 5c shows the chains of buildings around the boundary of the block. Figure 6 shows the final interpretation. It can be seen that the Blocks (1, 2, 3 and 4), misclassified by the previous algorithm, are interpreted correctly when spatial constraints are used.



Feature Extraction, Abstract, Figure 5 Steps of interpretation using the spatial constraints: **a** computing bounding rectangles, **b** principal directions, **c** finding building chains

Feature Extraction, Abstract, Figure 6 Result of interpretation using both scalar and spatial context constraints

Key Applications

Tactical Intelligence

In automated target recognition it is often critical to identify high value targets while reducing the risk of collateral damage. Depending on the mission, the goal is to minimize areas which are likely to have high population density. In such cases, it is important to identify the housing areas (single and multi-family) and shopping complexes during the high traffic times. Automated identification of such high-level features from information derived from remotely sensed imagery will likely be critical.

Urban Planning

Most of the urban centers in the world continue to see significant growth in population. Even smaller towns in rural areas near large metropolitan centers have seen growth in recent years. The growth in many cases is not coordinated with a master plan. Use of the approach described here, will allow the monitoring of the growth of these areas which is critical for planning and inventory.

Advanced Geo-Search Engines

Traditional text-based search engines are increasingly being augmented with geographic information. Satellite imagery is now available online for any part of the world, albeit with varying resolution. Extraction of abstract feature would allow the geo-searches to look for specific geographic features in satellite images. For example, an avid golfer may search for an abstract feature that consists of a golf course adjacent to an airport using the approach described here.

Crime Mapping

Law enforcement data in urban areas often show that some types of criminal activities are prevalent in a certain configuration of spatial features, such as, large downtown blocks with large parking lots and with easy access to a highway. One can define an abstract feature to capture this spatial configuration and identify it for better crime monitoring and alert.

Future Directions

As the amount of satellite–based and other remotely sensed imagery becomes even more available and methods to extract lower level features such as buildings and roads become more robust, there will be an increase in the interest to extract higher level features such as those described here. The focus is likely to change from deriving lower level to higher level feature extraction and from a structural description to a functional description of the scenes. Techniques that use ancillary data (e. g., census data, other GIS layers) to improve the efficiency and accuracy of feature extraction will become more feasible and critical.

Cross References

- ▶ Abstraction of GeoDatabases
- ▶ Crime Mapping and Analysis
- ▶ Geospatial Semantic Integration
- ▶ Patterns, Complex

Recommended Reading

1. Anderson, J.R., Hardy, E.E., Roach, J.T., Witmer, R.E.: A Land Use and Land Cover Classification System For Use With Remote Sensor Data. USGS Professional Paper **964** (1976)
2. Barzoharand, M., Cooper, D.B.: Automatic Finding of Main Roads in Aerial Images by Using Geometric Stochastic Models and Estimation. IEEE Transaction on Pattern. Anal. Mach. Intell. **18**(7), 707–721 (1996)
3. Fonseca, F., Egenhofer, M.: Ontology-Driven Geographic Information Systems. In: 7th ACM Symposium on Advances in Geographic Information Systems, Kansas City, MO, pp. 14–19. (1999)

4. Garry, S.: Ohio Land Use Land Cover Classification System. Remote Sensing Program Miscellaneous Report No. 17, Fifth Edition (1988)
5. Gruen, A., Li, H.: Semi-automatic Linear Feature Extraction by Dynamic Programming and LSB-Snake. Photogrammetric Eng. Remote Sensing **63**(8), 985–995 (1997)
6. Huertas, A., Lin, C., C., Nevatia, R.: Detection of Building form Monocular View of Aerial Scenes using Perceptual Grouping and Shadows. In: Proceeding DARPA Image Understanding Workshop, Washington, DC, pp. 253–260 (1993)
7. Li, Z., Yan, H., Ai, T., Chen, J.: Automated building generalization based on urban morphology and Gestalt theory. Int. J. Geogr. Inf. Sci. **18**(5), 513–534 (2004)
8. Lin, C., Nevatia, R.: Building Detection and Description from Monocular Aerial Images, Proceeding DARPA Image Understanding Workshop, Palm Springs, CA, pp. 461–468 (1996)
9. McKeown, D.M., Pane, J.F.: Alignment and Connection of Fragmented Linear Feature in Aerial Imagery. In: Proceedings of 1985 IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, pp. 55–61 (1985)
10. Wang, J., Zhang, Q.: Applicability of a Gradient Profile Algorithm for Road Network Extraction – Sensor Resolution and Background Consideration. Canadian J. Remote Sensing **26**(5), 428–439 (2000)

Feature Matching

► Conflation of Features

Features

► Geography Markup Language (GML)

Features, Linguistic

► Geospatial Semantic Integration

Features, Physical

► Geospatial Semantic Integration

FGDC

► Spatial Data Transfer Standard (SDTS)

Field Data

► Oracle Spatial, Raster Data

Filter and Refine Strategy

JORDAN WOOD
Department of Computer Science,
University of Minnesota, Minneapolis, MN, USA

Synonyms

Filter-Refine Paradigm

Definition

The filter and refine strategy is a general method for completing a computationally intensive task as quickly as possible. In some tasks, such as a spatial join, we need to do a computationally intensive step on only a small subset of the original data. In order to find this subset, we first apply a filter step which removes data that are not involved in our task. We then do the refining step on the remaining data.

Main Text

When we are using the filter and refine strategy with spatial data, we generally filter the data by running the algorithm in question on simplified representations of the spatial objects. Minimum bounding rectangles (MBRs) are commonly used for this purpose. The point of the filter step is not to reduce the data to only that data needed to find the answer to the problem, but rather to remove a significant part of the uninvolved data in a computationally efficient manner. For example, when trying to detect overlap present in a set of polygons, we might remove more data in the filter step if, for every polygon, we approximate using the smallest quadrilateral possible which contains the entire original polygon. However, if we impose the restriction that the quadrilateral must be a rectangle with sides which are parallel to the x and y axes, we can simplify both the algorithm which calculates the rectangles and the algorithm which performs the filter step.We therefore trade some computation in the refine step for potentially greater computation time savings in the filter step.

Cross References

► Indexing Spatial Constraint Databases
► Minimum Bounding Rectangle

Recommended Reading

1. Shekhar, S., Chawla, S.: Spatial Databases, A Tour, Pearson Education, Inc., New Jersey, USA (2003)
2. Orenstein, J.A., Spatial query processing in an object-oriented database system, Proceedings of the 1986 ACM SIGMOD international conference on Management data, p. 326–336, May 28–30, Washington, D.C., United States (1986)

Filtering

▶ Hierarchies and Level of Detail

Filter-Refine Paradigm

▶ Filter and Refine Strategy

FIPS 173

▶ Spatial Data Transfer Standard (SDTS)

FIPS PUB 173

▶ Spatial Data Transfer Standard (SDTS)

First Law of Geography

▶ Crime Mapping and Analysis

First-Order Logic with Constraints Queries

▶ Linear Versus Polynomial Constraint Databases

Fleet Management

▶ Routing Vehicles, Algorithms

Floating Car Data

DIETER PFOSER
RA Computer Technology Institute, Athens, Greece

Synonyms

Probe vehicle data; PVD; Vehicle tracking data

Definition

Floating car data (FCD) refers to using data generated by one vehicle as a sample to assess the overall traffic condition ("cork swimming in the river"). Typically this data comprises basic vehicle telemetry such as speed, direction and, most importantly, the position of the vehicle.

Main Text

The over-time-collected positional data component of FCD is referred to as *vehicle tracking data*. FCD can be obtained by tracking using either GPS devices or mobile phones. The latter type results in low-accuracy data. Depending on the collection method and data accuracy, more or less sophisticated map-matching algorithms have to be used to relate tracking data to a road network.

In database terms, the tracking data can be modeled in terms of a *trajectory*, which is obtained by interpolating the position samples. Typically, linear interpolation is used as opposed to other methods, such as polynomial splines. The sampled positions then become the endpoints of line segments and the movement of an object is represented by an entire polyline in 3D space.

FCD is a powerful means to assess traffic conditions in urban areas given that a large number of vehicles collect such data. Typical vehicle fleets comprise taxis, public transport vehicles, utility vehicles, but also private vehicles.

Cross References

▶ Dynamic Travel Time Maps
▶ Map-Matching

Recommended Reading

1. Schaefer, R.-P., Thiessenhusen, K.-U., Wagner, P.: A traffic information system by means of real-time floating-car data. In: Proc. ITS World Congress, Chicago (2002)

Flocking

▶ Movement Patterns in Spatio-temporal Data

Folksonomy

▶ Uncertainty, Semantic

Footprint

▶ Retrieval Algorithms, Spatial

Format

▶ Geography Markup Language (GML)

Fourier Series

► Constraint Databases and Data Interpolation

Four-Intersection Model

► Dimensionally Extended Nine-Intersection Model (DE-9IM)

Frame of Discernment

► Computing Fitness of Use of Geospatial Datasets

Free GIS

► MapWindow GIS

Frequent Itemset Discovery

MARKO SALMENKIVI
Department of Computer Science,
University of Helsinki, Helsinki, Finland

Synonyms

Market-basket analysis

Definition

Consider the set of all products sold by a supermarket. Assume that the owner of the supermarker is interested in finding out subsets of products that are often purchased together. Each customer transaction is stored in a transaction database, indicating the products that the customer purchased together. The database can be described as a table, whose columns are the products (items), and the rows are the transactions. The value of a specific entry, that is, (row, column)-pair, in the table is 1 if the corresponding product was purchased in the transaction, and 0 otherwise. The task is to find itemsets such that the items frequently occur in the same row (products purchased together). The most important interestingness measure in frequent itemset mining is *support* of an itemset. It is defined as the fraction of rows of the database that contain all the items $x \in X$. An itemset is *frequent* if its support exceeds a user-specified threshold value.

Association rules are a closely related pattern class. Let R be a set of products, r a transaction database, and $X, Y \subseteq R$ itemsets. Then $X \rightarrow Y$ is an association rule over r. The interestingness of an association rule $X \rightarrow Y$ is usually measured by its support defined as $support(X \rightarrow Y) = support(X \cup Y)$, and confidence: $conf(X \rightarrow Y) = \frac{support(X \cup Y, r)}{support(X, r)}$. Thus, the confidence is the conditional probability that a randomly chosen row from r that contain the items in X also contain the items in Y. Given thresholds for support and confidence, the association rule mining task in a given database is to find all the rules, whose supports and confidences exceed the thresholds.

Main Text

One of the first significant contributions of data mining research was the notion of association rule. All interesting association rules from a relational database can be found by solving a subtask known as frequent itemset discovery, often called market-basket analysis.

The name market-basket analysis originates with the domain of the analysis of customer behavior in supermarkets, which was the starting point of the methodology in the late 1980's. The most well-known algorithm for finding all frequent itemsets is Apriori introduced in 1994 [1]. Many algorithms that are more efficient than Apriori have been introduced later (e. g., [4]). It was soon realized that frequent itemset discovery often resulted in very large number of frequent itemsets and confident association rules. Recent research has concentrated on developing condensed representations (closed sets, non-derivable sets, non-derivable association rules) for the set of interesting patterns. The basic idea is that usually in practice a large part of the interesting patterns are redundant in the sense that they can be deduced from a subset of all frequent patterns [2,3]. This leads to a new interestingness measure: a pattern is interesting only if its support (and/or confidence) cannot be derived from the supports (and/or confidences) of its subpatterns.

Cross References

► Co-location Pattern
► Co-location Pattern Discovery
► Frequent Pattern

Recommended Reading

1. Agrawal, R., Ramakrishnan, S.: Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases, pp. 487–499, September 12–15 (1994)
2. Goethals, B., Muhonen, J., Toivonen, H.: Mining non-derivable association rules. SIAM International Conference on Data Mining, 239–249, Newport Beach, California (2005)

3. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Discovering frequent closed itemsets for association rules. In: Proceedings of the International Conference of Database Theory – ICDT '99. Lecture Notes in Computer Science vol. 1540, pp. 398–416 (1999)
4. Zaki, M.: Scalable algorithms for association mining. IEEE Trans Knowl Data Eng **12**(3), 372–390 (2000)

Frequent Itemset Mining

► Patterns, Complex

Frequent Pattern

MARKO SALMENKIVI
Department of Computer Science,
University of Helsinki, Helsinki, Finland

Synonyms

Interesting pattern; Selected pattern

Definition

Given a set of patterns, i. e., a class of expressions about databases, and a predicate to evaluate whether a database satisfies a pattern, the frequent pattern mining task is to determine which patterns are satisfied by a given database. Formally, assume that $\mathcal{P}$ is a set and q is a predicate p : $\mathcal{P} \times \{\mathbf{r} \mid \mathbf{r}$ is a database$\} \rightarrow \{$true, false$\}$. Elements of $\mathcal{P}$ are called patterns, and q is a selection criterion over $\mathcal{P}$. Given a pattern φ in $\mathcal{P}$ and a database $\mathbf{r}$, φ is selected if $q(\varphi, \mathbf{r})$ is true. Given a database $\mathbf{r}$, the theory $\mathcal{T}/\mathcal{P}, \mathbf{r}, q)$ of $\mathbf{r}$ with respect to $\mathcal{P}$ and q is $\mathcal{T}/\mathcal{P}, \mathbf{r}, q) = \{\phi \in \mathcal{P} | q(\phi, \mathbf{r})$is true$\}$.

Main Text

It is important to note that the interpretation of "$q(\varphi,\mathbf{r})$ is true" depends on the application domain. For some applications it could mean that φ occurs often enough in $\mathbf{r}$, that φ is true or almost true in $\mathbf{r}$, or that φ defines, in some way, an interesting property of subgroup of $\mathbf{r}$.

For computational purposes, a very important property of the selection criterion (i. e., interestingness measure) is *monotonicity*. Let $\preceq$ be a partial order on the patterns in $\mathcal{P}$. If for all databases $\mathbf{r}$ and patterns $\varphi, \theta \in \mathcal{P}$ it holds that $q(\varphi, r)$ and $\theta \preceq \varphi$ imply $q(\theta, \mathbf{r})$, then $\preceq$ is a *specialization relation* on $\mathcal{P}$ with respect to q. Then q is monotonous with respect to the pattern class and the specialization relation. If the monotonicity holds, levelwise algorithms (such as Apriori-algorithm [1]) can be used to find all interesting patterns.

For instance, in market-basket analysis (frequent itemset discovery) a natural way to define the specialization relation is the size of the itemset. Here the number of items in itemset φ is denoted by $|\varphi|$, and $\theta \preceq \varphi$ if and only if $|\theta| \geq |\varphi|$. Clearly, the support of φ is monotonous with respect to the size of the pattern. Similarly, in the case of the co-location patterns, the definition of specialization relation can be based on the number of features in the patterns. Participation ratio and prevalence, for instance, are monotonous with respect to the specialization relation.

F

Cross References

► Co-location Pattern
► Co-location Pattern Discovery
► Co-location Patterns, Interestingness Measures
► Frequent Itemset Discovery

Recommended Reading

1. Agrawal, R., Ramakrishnan, S.: Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases, pp. 487–499, September 12–15 (1994)
2. Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Mining Knowl Disc **1**(3), 241–258 (1997)

Functional Description

► Feature Extraction, Abstract

Fundamental Matrix

► Photogrammetric Methods

Fuzzy Sets

► Objects with Broad Boundaries

G/Technology

► Intergraph: Real Time Operational Geospatial Applications

Gaussian

► Hurricane Wind Fields, Multivariate Modeling

Gaussian Process Models in Spatial Data Mining

NAREN RAMAKRISHNAN[1], CHRIS BAILEY-KELLOGG[2]
[1] Department of Computer Science, Virginia Tech, Blacksburg, VA, USA
[2] Department of Computer Science, Dartmouth College, Hanover, NH, USA

Synonyms

Active data mining

Definition

Gaussian processes (GPs) are local approximation techniques that model spatial data by placing (and updating) priors on the covariance structures underlying the data. Originally developed for geo-spatial contexts, they are also applicable in general contexts that involve computing and modeling with multi-level spatial aggregates, e. g., modeling a configuration space for crystallographic design, casting folding energies as a function of a protein's contact map, and formulation of vaccination policies taking into account social dynamics of individuals. Typically, we assume a parametrized covariance structure underlying the data to be modeled. We estimate the covariance parameters conditional on the locations for which we have observed data, and use the inferred structure to make predictions at new locations. GPs have a probabilistic basis that allow us to estimate variances at unsampled locations, aiding in the design of targeted sampling strategies.

Historical Background

The underlying ideas behind GPs can be traced back to the geostatistics technique called kriging [4], named after the South African miner Danie Krige. Kriging in this literature was used to model response variables (e. g., ozone concentrations) over 2D spatial fields as realizations of a stochastic process. Sacks et al. [12] described the use of kriging to model (deterministic) computer experiments. It took more than a decade from this point for the larger computer science community to investigate GPs for pattern analysis purposes. Thus, in the recent past, GPs have witnessed a revival primarily due to work in the statistical pattern recognition community [5] and graphical models literature [3]. Neal established the connection between Gaussian processes and neural networks with an infinite number of hidden units [8]. Such relationships allow us to take traditional learning techniques and re-express them as imposing a particular covariance structure on the joint distribution of inputs. For instance, we can take a trained neural network and mine the covariance structure implied by the weights (given mild assumptions such as a Gaussian prior over the weight space). Williams motivates the usefulness of such studies and describes common covariance functions [14]. Williams and Barber [15] describe how the Gaussian process framework can be extended to classification in which the modeled variable is categorical. Since these publications were introduced, interest in GPs has exploded with rapid publications in conferences such as ICML, NIPS; see also the recently published book by Rasmussen and Williams [11].

Scientific Fundamentals

A GP can be formally defined as a collection of random variables, any finite subset of which have a (multivariate) normal distribution. For simplicity, we assume 2D spatially distributed (scalar) response variables t_i, one for each location $\mathbf{x_i} = [x_{i1}, x_{i2}]$ where we have collected a data sample. Observe that, in the limiting case, each random variable has a Gaussian distribution (but it is not true that any collection of Gaussian random variables will induce a GP). Given a dataset $\mathcal{D} = \{\mathbf{x_i}, t_i\}$, $i = 1 \ldots n$, and a new data point $\mathbf{x_{n+1}}$, a GP can be used to model the posterior $P(t_{n+1} \mid \mathcal{D}, x_{n+1})$ (which would also be a Gaussian). This is essentially what many Bayesian modeling techniques do (e. g., least squares approximation with normally distributed noise), however, it is the specifics of how the posterior is modeled that make GPs distinct as a class of modeling techniques.

To make a prediction of t_{n+1} at a point $\mathbf{x_{n+1}}$, GPs place greater reliance on t_i's from nearby points. This reliance is specified in the form of a covariance prior for the process. One example of a covariance prior is:

$$\mathrm{Cov}(t_i, t_j) = \alpha \exp\left(-\frac{1}{2}\sum_{k=1}^{2} a_k (x_{ik} - x_{jk})^2\right) . \tag{1}$$

Intuitively, this function captures the notion that response variables at nearby points must have high correlation. In Eq. 1, α is an overall scaling term, whereas a_1, a_2 define the length scales for the two dimensions. However, this prior (or even its posterior) does not directly allow us to determine t_j from t_i since the structure only captures the covariance; predictions of a response variable for new sample locations are thus conditionally dependent on the measured response variables and *their* sample locations. Hence, we must first estimate the covariance parameters (a_1, a_2, and α) from $\mathcal{D}$, and then use these parameters *along with* $\mathcal{D}$ to predict t_{n+1} at $\mathbf{x_{n+1}}$.

Before covering the learning procedure for the covariance parameters (a_1, a_2, and α), it is helpful to develop expressions for the posterior of the response variable in terms of these parameters. Since the jpdf of the response variables $P(t_1, t_2, \ldots, t_{n+1})$ is modeled Gaussian (we will assume a mean of zero), we can write:

$$P(t_1, t_2, \ldots, t_{n+1} \mid \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_{n+1}}, \mathrm{Cov}_{n+1}) = \frac{1}{\lambda_1} \cdot \exp\left(-\frac{1}{2}\,[t_1, t_2, \ldots, t_{n+1}]\,\mathrm{Cov}_{n+1}^{-1}\,[t_1, t_2, \ldots, t_{n+1}]^{\mathrm{T}}\right)$$

where we ignore λ_1 as it is simply a normalizing factor. Here, Cov_{n+1} is the covariance matrix formed from the $(n+1)$ data values $(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_{n+1}})$. A distribution for the unknown variable t_{n+1} can then be obtained as:

$$\begin{aligned} &P(t_{n+1} | t_1, t_2, \ldots, t_n, \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_{n+1}}, \mathrm{Cov}_{n+1}) \\ &= \frac{P(t_1, t_2, \ldots, t_{n+1} \mid \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_{n+1}}, \mathrm{Cov}_{n+1})}{P(t_1, t_2, \ldots, t_n \mid \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_{n+1}}, \mathrm{Cov}_{n+1})} \\ &= \frac{P(t_1, t_2, \ldots, t_{n+1} \mid \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_{n+1}}, \mathrm{Cov}_{n+1})}{P(t_1, t_2, \ldots, t_n \mid \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}, \mathrm{Cov}_n)} , \end{aligned}$$

where the last step follows by conditional independence of $\{t_1, t_2, \ldots, t_n\}$ w.r.t. $\mathbf{x_{n+1}}$ and the part of Cov_{n+1} not contained in Cov_n. The denominator in the above expression is another Gaussian random variable given by:

$$\begin{aligned} &P(t_1, t_2, \ldots, t_n \mid \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}, \mathrm{Cov}_n) \\ &= \frac{1}{\lambda_2} \exp\left(-\frac{1}{2}\,[t_1, t_2, \ldots, t_n]\,\mathrm{Cov}_n^{-1}\,[t_1, t_2, \ldots, t_n]^{\mathrm{T}}\right) . \end{aligned}$$

Putting it all together, we get:

$$\begin{aligned} &P(t_{n+1} | t_1, t_2, \ldots, t_n, \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_{n+1}}, \mathrm{Cov}_{n+1}) \\ &= \frac{\lambda_2}{\lambda_1} \exp\Bigg(-\frac{1}{2}[t_1, t_2, \ldots, t_{n+1}]\,\mathrm{Cov}_{n+1}^{-1}[t_1, t_2, \ldots, t_{n+1}]^{\mathrm{T}} \\ &\quad - \frac{1}{2}[t_1, t_2, \ldots, t_n]\,\mathrm{Cov}_n^{-1}\,[t_1, t_2, \ldots, t_n]^{\mathrm{T}}\Bigg) . \end{aligned}$$

Computing the mean and variance of this Gaussian distribution, we get an estimate of t_{n+1} as:

$$\hat{t}_{n+1} = \mathbf{k}^{\mathrm{T}} \mathrm{Cov}_n^{-1} [t_1, t_2, \ldots, t_n] , \tag{2}$$

and our uncertainty in this estimates as:

$$\sigma^2_{\hat{t}_{n+1}} = k - \mathbf{k}^{\mathrm{T}} \mathrm{Cov}_n^{-1} \mathbf{k} , \tag{3}$$

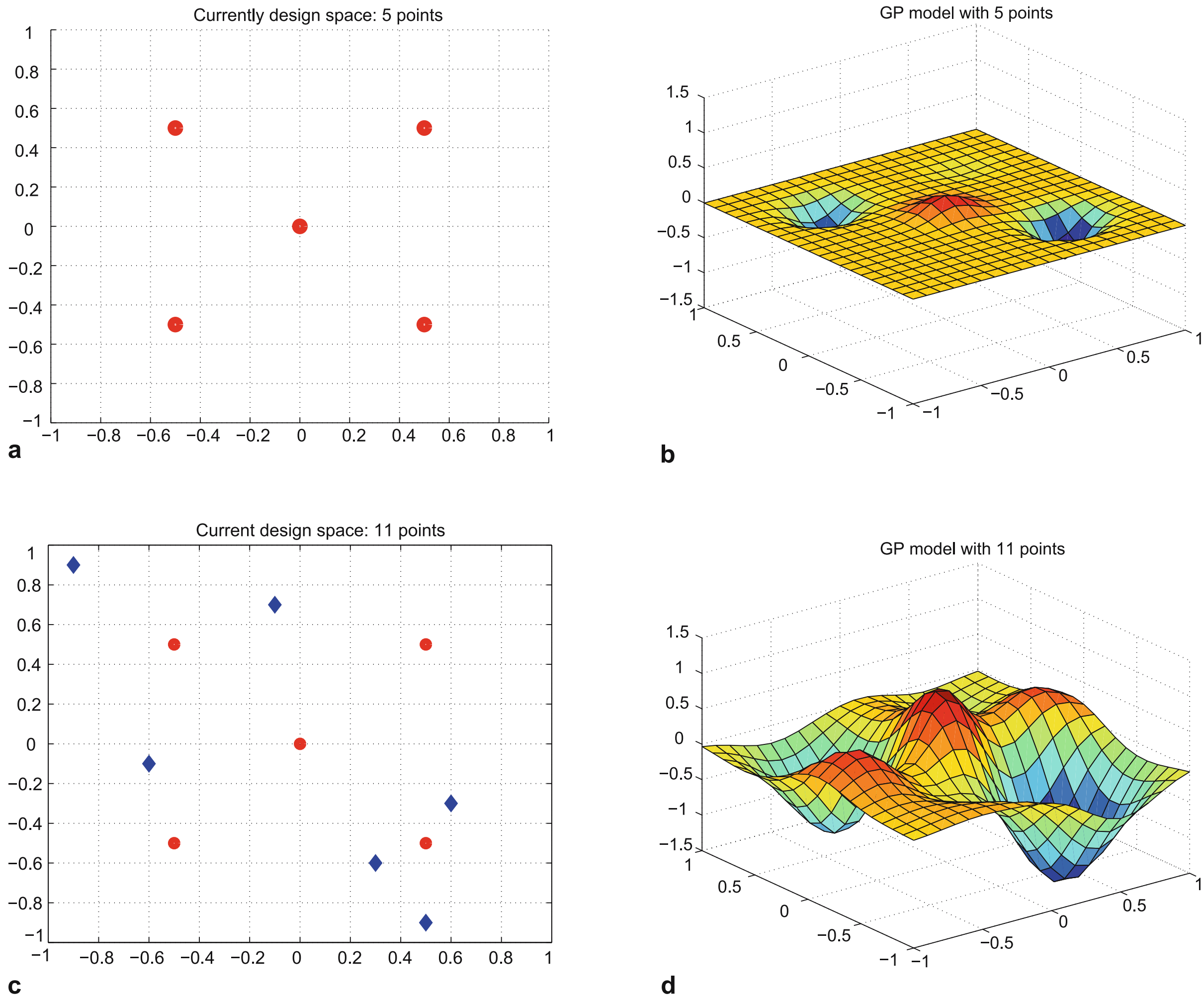
where $\mathbf{k}^{\mathrm{T}}$ represents the n-vector of covariances with the new data point:

$$\mathbf{k}^{\mathrm{T}} = [\mathrm{Cov}(\mathbf{x_1}, \mathbf{x_{n+1}})\ \mathrm{Cov}(\mathbf{x_2}, \mathbf{x_{n+1}}) \ldots \mathrm{Cov}(\mathbf{x_n}, \mathbf{x_{n+1}})] ,$$

and k is the $(n+1,\ n+1)$ entry of Cov_{n+1}. Equations 2 and 3, together, give us both an approximation at any given point and an uncertainty in this approximation; they will serve as the basic building blocks for closing-the-loop between data modeling and higher level mining functionality.

The above expressions can be alternatively derived by positing a linear probabilistic model and optimizing for the MSE (mean squared error) between observed and predicted response values (e. g., see [12]). In this sense, the Gaussian process model considered here is also known as the BLUE (best linear unbiased estimator), but GPs are not restricted to linear combinations of basis functions.

To apply GP modeling to a given dataset, one must first ensure that the chosen covariance structure matches the data characteristics. The above example used a stationary structure which applies when the covariance is translation invariant. Various other functions have been studied in the literature (e. g., see [7,9,12]), all of which satisfy the required property of positive definiteness of a covariance



Gaussian Process Models in Spatial Data Mining, Figure 1 Active mining with Gaussian processes. An initial sample of data points (**a**; shown as *red circles*) gives a preliminary approximation to the target function (**b**). Active sampling suggests new locations (**c**; *blue diamonds*) that improve the quality of approximation (**d**)

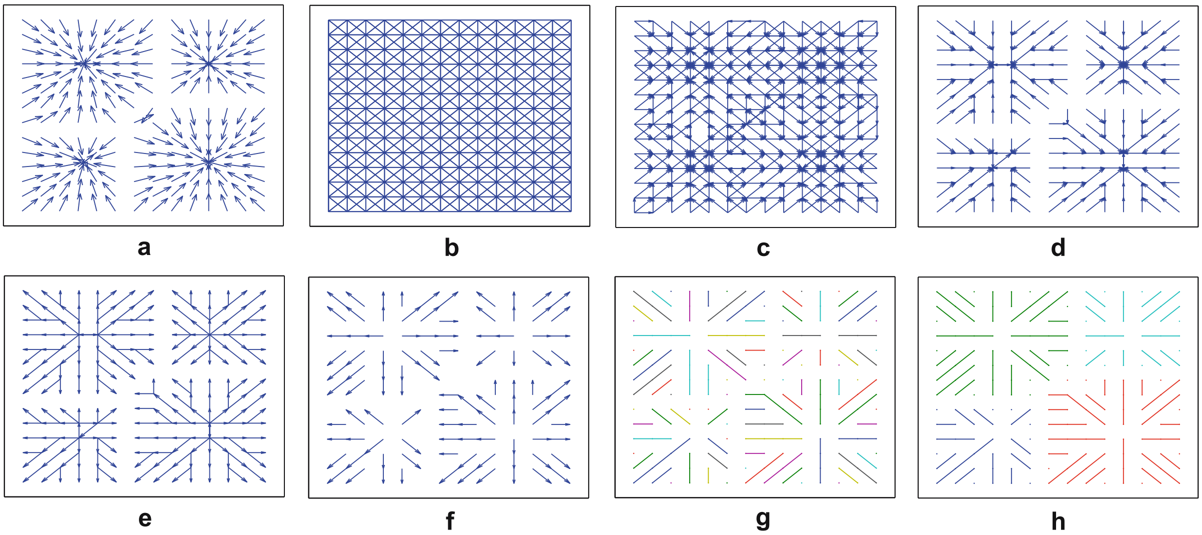
matrix. The simplest covariance function yields a diagonal matrix, but this means that no data sample can have an influence on other locations, and the GP approach offers no particular advantages. In general, by placing a prior directly on the function space, GPs are appropriate for modeling 'smooth' functions. The terms a_1, a_2 capture how quickly the influence of a data sample decays in each direction and, thus, the length scales for smoothness.

An important point to note is that even though the GP realization is one of a random process, we can nevertheless build a GP model for deterministic functions by choosing a covariance structure that ensures the diagonal correlations to be 1 (i. e., perfect reproducibility when queried for a sample whose value is known). Also, the assumption of zero mean for the Gaussian process can be relaxed by including a constant term (gives another parameter to be estimated) in the covariance formulation.

Learning the GP parameters $\theta = (a_1, a_2, \alpha)$ can be undertaken in the maximum likelihood (ML) and maximum a posteriori (MAP) frameworks, or in the true Bayesian setting where we obtain a distribution over values. The log-likelihood for the parameters is given by:

$$\begin{aligned}\mathcal{L} &= \log P(t_1, t_2, \ldots, t_n | \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}, \theta) \\ &= c + \log P(\theta) - \frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\operatorname{Cov}_n| \\ &\quad - \frac{1}{2}[t_1, t_2, \ldots, t_n]\operatorname{Cov}_n^{-1}[t_1, t_2, \ldots, t_n]^{\mathrm{T}}.\end{aligned}$$

To optimize for the parameters, we can compute partial derivatives of the log-likelihood for use with a conjugate



Gaussian Process Models in Spatial Data Mining, Figure 2 Computation of multi-level spatial aggregations. **a** Input vector field. **b** 8-adjacency neighborhood graph. **c** Forward neighbors. **d** Best forward neighbors. **e** Neighborhood graph transposed from best forward neighbors. **f** Best backward neighbors. **g** Resulting adjacencies redescribed as curves. **h** Higher-level aggregation and classification of curves whose flows converge

gradient or other optimization algorithm:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \log P(\theta)}{\partial \theta} - \frac{1}{2}\operatorname{tr}\left(\mathrm{Cov}_n^{-1}\frac{\partial\,\mathrm{Cov}_n^{-1}}{\partial \theta}\right) + \frac{1}{2}[t_1, t_2, \ldots, t_n]\,\mathrm{Cov}_n^{-1}\frac{\partial\,\mathrm{Cov}_n^{-1}}{\partial \theta}\,\mathrm{Cov}_n^{-1}\,[t_1, t_2, \ldots, t_n]^{\mathrm{T}},$$

where $\mathrm{tr}(\cdot)$ denotes the trace function. In our running example, we need only estimate three parameters for θ, well within the purview of modern numerical optimization software. For larger numbers of parameters, we can resort to the use of Monte Carlo Markov Chain (MCMC) methods [9].

Key Applications

Gaussian processes are applicable for spatial modeling tasks in a variety of application contexts.

Active Data Mining

In applications such as crystallographic design, where one must characterize a configuration space or design space in terms of spatial aggregates, data collection can become costly. In these applications, it is beneficial to collect data only at those locations that are deemed important to support a data mining objective. Toward this goal, we can use GPs to work with only a sparse set of samples and, based on the quality of approximation, provide objective criteria for choosing the next sample point. Figure 1 depicts a 2D example of 'seeding' a GP with an initial sample of data points (left two frames), thereby defining functionals over the unsampled region (not shown) which are then optimized to arrive at new locations to sample (right two frames).

Geostatistical Motion Interpolation

Gaussian processes have been used to solve the motion interpolation or 'in-betweening' task in computer graphics [6]. Given two frames denoting an individual in motion and a multi-parameter space of control variables, a GP model synthesizes smooth animations that emulate natural human movements and obey geographical constraints. GPs have also been used for robotic imitation by modeling data gathered from human motion capture devices [13].

Spatial Aggregation

GPs can be used to model the multi-layer construction of spatial aggregates from data. Figure 2 describes steps in aggregating individual vectors, first into streamlines and then into convergent flows, using a custom spatial aggregation algorithm. The qualitative nature of such aggregations can be summarized computationally using GPs to yield mathematical models of data mining algorithms.

Sensor Networks

GPs have been applied in sensor network contexts [2], e. g., monitoring physical variables over an environment using a number of sensing devices. By parametrizing the covariance distribution of the physical variable and determining where uncertainty of estimation is highest, one can design judicious sensor placement policies.

Future Directions

There are many open and promising directions for Gaussian processes research. There are new, overlapping, notions of spatiality that must be modeled in applications such as pandemic disease modeling [1]. In these contexts, the definition of nearby random variables is drawn both from geographical distance as well as social proximity considerations. From work that merely estimates parameters of covariance functions, new work has begun to learn the structure of covariance functions. These will undoubtedly become more critical as new applications of GPs are explored. Finally, as the sensor network application reveals, the development of new objective functions for active data mining is crucial, especially for those that are suited for distributed model building.

Cross References

► Kriging

Acknowledgments

The figures in this chapter were published previously in [10] and reproduced here with permission from SIAM Press.

Recommended Reading

1. Bailey-Kellogg, C., Ramakrishnan, N., Marathe, M.V.: Spatial Data Mining for Pandemic Preparedness. ACM SIGKDD Explorations **8**(1), 80–82 (2006)
2. Guestrin, C., Krause, A., Singh, A.P.: Near-optimal Sensor Placements in Gaussian Processes. In: Proceedings of the Twenty-Second International Conference (ICML 2005), pp. 265–272. (2005)
3. Jordan, M.I.(ed.): Learning in Graphical Models. MIT Press, Cambrige, MA (1998)
4. Journel, A.G., Huijbregts, C.J.: Mining Geostatistics. Academic Press, New York (1992)
5. MacKay, D.J.: Gaussian Processes: A Replacement for Supervised Neural Networks? In: Lecture Notes of Tutorial at Neural Information Processing Systems (NIPS'97) (1997)
6. Mukai, T., Kuriyama, S.: Geostatistical Motion Interpolation. ACM Trans. Graph. **24**(3), 1062–1070 (2005)
7. Nabney, I.T.: Netlab: Algorithms for Pattern Recognition. Springer-Verlag, New York (2002)
8. Neal, R.M.: Bayesian Learning for Neural Networks. Lecture Notes in Statistics No. 118. Springer-Verlag, New York (1996)
9. Neal, R.M.: Monte Carlo Implementations of Gaussian Process Models for Bayesian Regression and Classification. Technical Report 9702, Department of Statistics, University of Toronto, January 1997
10. Ramakrishnan, N., Bailey-Kellogg, C., Tadepalli, S., Pandey, V.N.: Gaussian Process for Active Data Mining of Spatial Aggregates. In: Proceedings of the SIAM International Conference on Data Mining (2005)
11. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambrige, MA (2006)
12. Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and Analysis of Computer Experiments. Stat. Sci. **4**(4), 409–435 (1989)
13. Shon, A.P., Grochow, K., Rao, R.P.N.: Robotic Imitation from Human Motion Capture using Gaussian Processes. In: Technical Report, Department of Computer Science and Engineering, University of Washington, Seattle, WA (2006)
14. Williams, C.K.I.: Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond. In: Jordan, M.I.(ed.) Learning in Graphical Models, pp. 599–621. MIT Press, Cambridge, MA (1998)
15. Williams, C.K.I., Barber, D.: Bayesian Classification with Gaussian Processes. IEEE PAMI **20**(12), 1342–1351 (1998)

G

Gazeteer

► Retrieval Algorithms, Spatial

GDAL

► Open-Source GIS Libraries

GE Smallworld

► Smallworld Software Suite

Geary Coefficient

► Geary's *C*

Geary Ratio

► Geary's *C*

Geary's *C*

XIAOBO ZHOU, HENRY LIN
Department of Crop and Soil Sciences, The Pennsylvania State University, University Park, PA, USA

Synonyms

Geary's index; Geary ratio; Geary coefficient

Definition

Geary's C tests statistics for spatial autocorrelation by using the sum of squared differences between pairs of data of variable x as a measure of covariation

$$C = \frac{(n-1)\sum_i \sum_j w_{ij}(x_i - x_j)^2}{2nS^2 \sum_i \sum_j w_{ij}} .$$

Where x_i denotes the observed value at location i,

$$S^2 = \frac{1}{n}\sum_i (x_i - \bar{x})^2 ,$$

$\bar{x}$ is the mean of the variable x over the n locations and w_{ij} are the elements of the spatial weights matrix, defined as 1 if location i is contiguous to location j and 0 otherwise. Other spatial weights matrices can also be used.

Main Text

Geary's C ranges from 0 to a positive value. The value of C is 1 in the absence of spatial autocorrelation. A low value of C $(0 < C < 1)$ represents a positive spatial autocorrelation and approaches zero for strong autocorrelation. A high value $(C > 1)$ represents negative spatial autocorrelation with greater values corresponding to a strong negative spatial autocorrelation. Geary's C is more sensitive to the variation of neighborhoods than to the global variation.

Cross References

► Autocorrelation, Spatial

Geary's Index

► Geary's *C*

Generalization

► Map Generalization
► Privacy Threats in Location-Based Services

Generalization and Symbolization

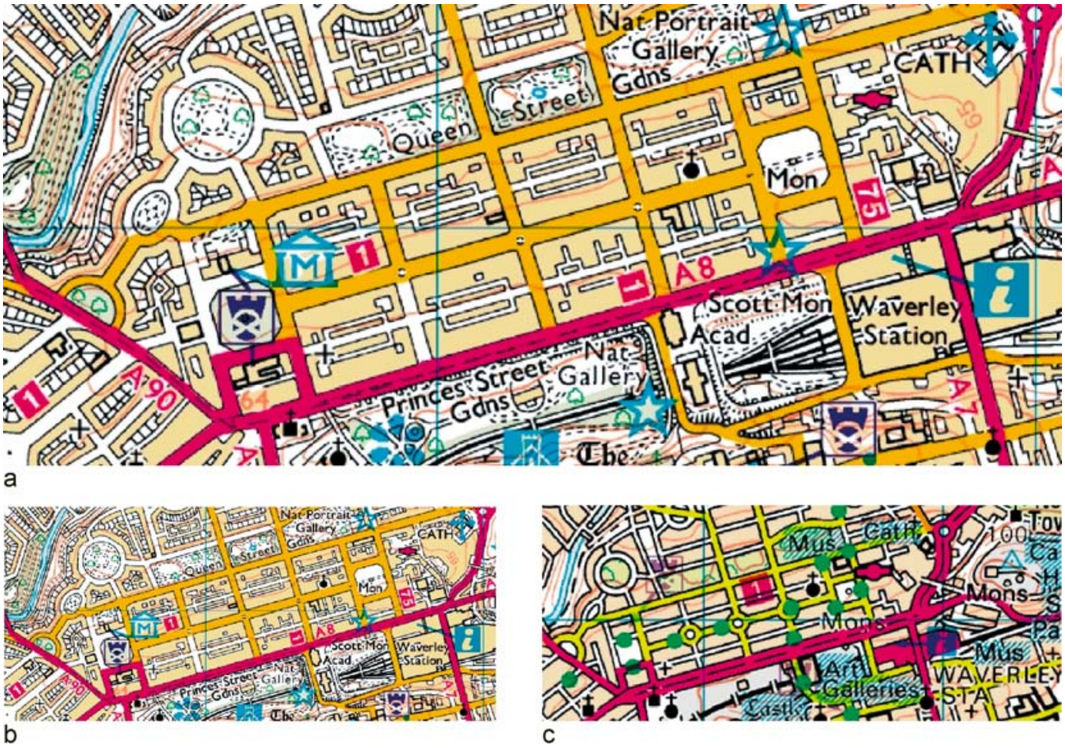
William A. Mackaness, Omair Chaudhry
Institute of Geography, School of GeoSciences,
The University of Edinburgh, Edinburgh, UK

Definition

Map generalization is a process concerned with the application of a set of algorithms to geographic data (represented in vector form) in order to control the optimal representation of geographic phenomenon at a range of different scales or levels of detail. In that sense, generalization seeks to mirror the process of map design previously undertaken by the human cartographer. In the context of geographical information systems (GIS), this process is modeled as two sets of operations: the first is a set of database operations (model generalization) and the second is a set of visualization operations (cartographic generalization). Model generalization is concerned with simplifying the representational form in order to achieve efficiencies in data storage, selecting classes of objects according to some specified scale and map theme, and aggregating groups of objects in accordance with scale constraints. Cartographic generalization (a compliment to model generalization) is concerned with the optimal portrayal of those selected and aggregated features. Cartographic generalization involves selecting appropriate symbols, giving emphasis to some of the feature's defining characteristics, and where there are dense regions of features, omitting some features or making small displacements to features in order to resolve ambiguity. Figure 1 seeks to demonstrate the need for generalization. Simple photographic reduction is not sufficient (Fig. 1b); thus the aim of map generalization is to derive smaller scale mapping (Fig. 1c) from detailed, large scale mapping (Fig. 1a).

Historical Background

All geographical processes are imbued with scale [1:214], thus issues of scale are an essential consideration in geographical problem solving. The scale of observation governs what phenomena can be viewed, what patterns are discernible, and what processes can be inferred. Study in the geosciences is focused both on the detail of those phenomena, as well as the broad linkages across regional and global space. Choosing scales of analysis, comparing output at different scales, describing constructions of scale [2] are all common practices in the geosciences. Traditionally it has been the cartographer's responsibility to select a scale, to symbolize the phenomena, and to give meaning through the addition of appropriate contextual information. The paper map was the basis of geographical inquiry. Indeed it was argued that if the problem 'cannot be studied fundamentally by maps – usually by a comparison of several maps – then it is questionable whether or not it is within the field of geography' [3:249]. Information technology has not devalued the power of the map, but it has driven a series of paradigm shifts in the storage, representation and interaction with geographical information. Early work in automated mapping focused on supporting the activities of the human cartographer who remained central to the map design process. Current research is focused more on ideas of autonomous design – systems capable of selecting optimum solutions among a variety of candidate



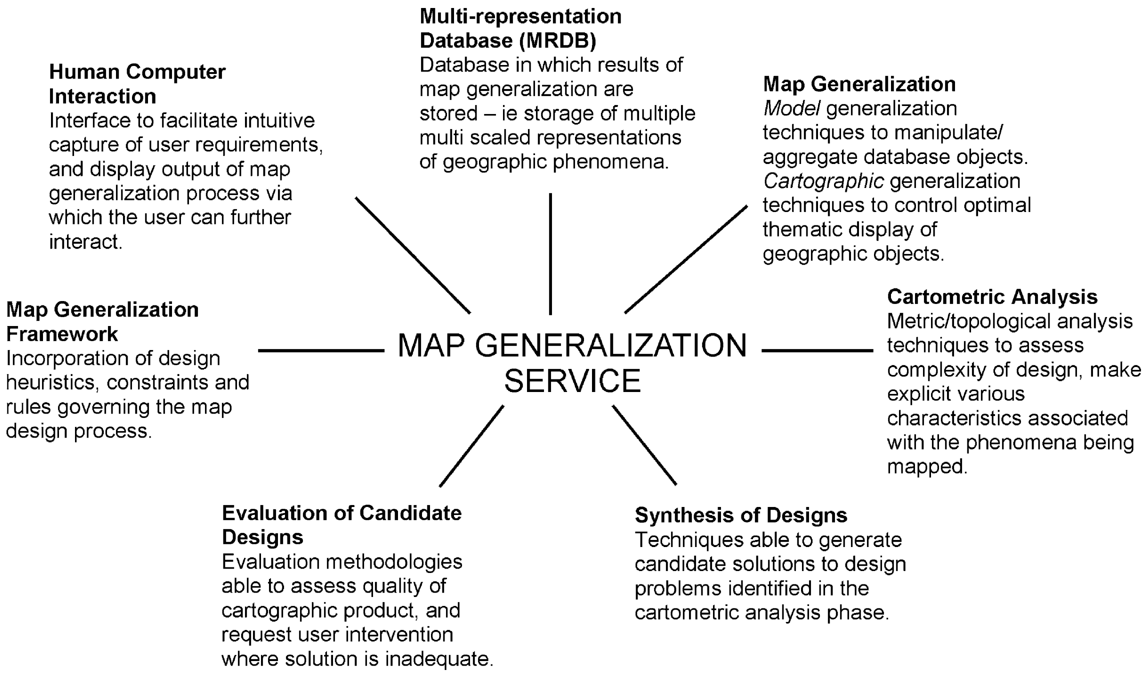
Generalization and Symbolization, Figure 1 Map generalization – creating different geographies of space (Mapping is Ordnance Survey © Crown Copyright. All rights reserved)

solutions delivered over the web, in a variety of thematic forms, in anticipation of users who have little or no cartographic skill. Historically the paper map reflected a state of knowledge. Now it is the database that is the knowledge store, with the map as the metaphorical window by which geographic information is dynamically explored. In these interactive environments, the art and science of cartography is being extended to support the integration of distributed data collected at varying levels of detail, whilst conforming to issues of data quality and interoperability. With respect to map generalization, the challenge is in developing a set of algorithms and methodologies that mirror the service traditionally provided by the human cartographer, yet takes advantage of the paradigm shift afforded by information science in interacting with, and exploring geographic information.

Scientific Fundamentals

The human cartographer provides a service that involves interpreting the requirements of the user, creating and executing a design to a very high quality and clarity according to a theme and scale, and one that is void of ambiguity. Over the past thirty years huge advances in database technology, together with developments in geo-visualization [4,5] and interactive and web based mapping has disrupted and further displaced the role of the cartographer. The digital map now acts as a window by which to search and explore the underlying database, and the cartographer has supposedly been replaced by symbol libraries and color ramps that may, in the absence of cartographic expertise, facilitate 'the creation of cartographic monstrosities with unprecedented ease' [6].

Within this paradigm shift, the requirement to view the world at different scales (or multiple levels of detail) has remained, as has the requirement to produce high quality cartographic products. Initially paper maps at different scales were digitized and stored in different databases. However there is huge redundancy in this model as changes in the real world have to be reflected in changes in each of the databases. A new line of thinking has emerged which asks whether it is possible to store the phenomenon once (at a very high level of detail), and then apply a range of algorithms in order to control the selection and representation of the phenomenon in a form appropriate to the intended scale. There are significant benefits to this line of thinking; maintaining a single database is more cost effective than maintaining multiple databases; a high level of consistency can be maintained between different datasets; duplication of storage can be avoided thus obviating the



Generalization and Symbolization, Figure 2 The components of a Map Generalization Service

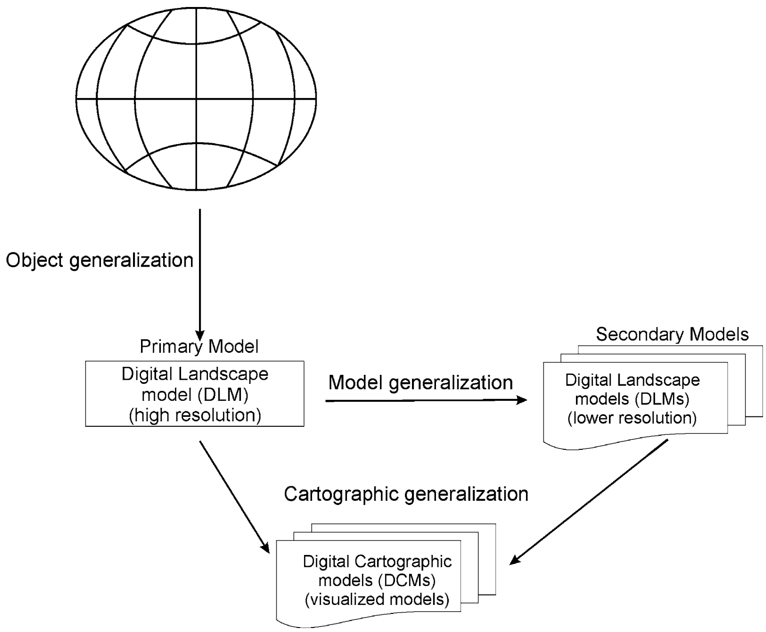
need to make multiple updates across separate databases each time a change occurs in the real world. Most importantly it offers the opportunity to share data, enabling integration of data from disparate sources, captured at different levels of detail.

These benefits are premised on the existence of a set of algorithms that can, with minimum intervention from the user, control the selection and representation of geographic phenomenon according to a specified scale and theme. The science of 'map generalization' is all about designing such algorithms; algorithms that manipulate and symbolize the geometric primitives stored in the database. Map generalization can also be viewed as a service that anticipates users unfamiliar with cartographic concepts, and with poor evaluation skills. Such a service must contain the following components: a database capable of storing multiple representations of geographic phenomena, a set of model and cartographic generalization techniques to create such multiple representations, and design heuristics that govern the appropriate choice and sequencing of generalization techniques. The evaluation of any candidate design requires the system to create alternate candidate designs (synthesis), and to evaluate and select the best solution (which in turn requires a set of cartometric analysis tools). Interpreting the map requirements of the user, and presenting solutions in response requires an interface that can 'translate' straightforward requests into rich specifications and parameter setting. These are deemed to be the essential components of a Map Generalization Service (Fig. 2).

This chapter begins by describing the techniques used to manipulate objects within the database. It then describes some of the frameworks designed to support their application in the overall design of the map. The discussion that follows this, argues that high levels of automation can only be achieved if the automated environment includes methods of evaluation. The entry concludes with a brief discussion of the changing context of map generalization within developing applications (such as exploratory data analysis and location based services).

Tools and Techniques for Map Generalization

The goal of map generalization is to give emphasis to salient objects and their properties whilst omitting less important qualities with respect to the scale and the purpose of a map. Therefore a system is needed that supports manipulation of map objects and their relationships, and more generally supports the representation of phenomena at different scales. For example at the finest scale each individual building, street light and pavement might be represented. But at a coarse scale, all of this might be subsumed by a single 'dot' (with say, the word 'London' next to it), representing the idea of 'city' in which all those buildings are contained. Therefore the requirements for a map generalization system are: 1) a database



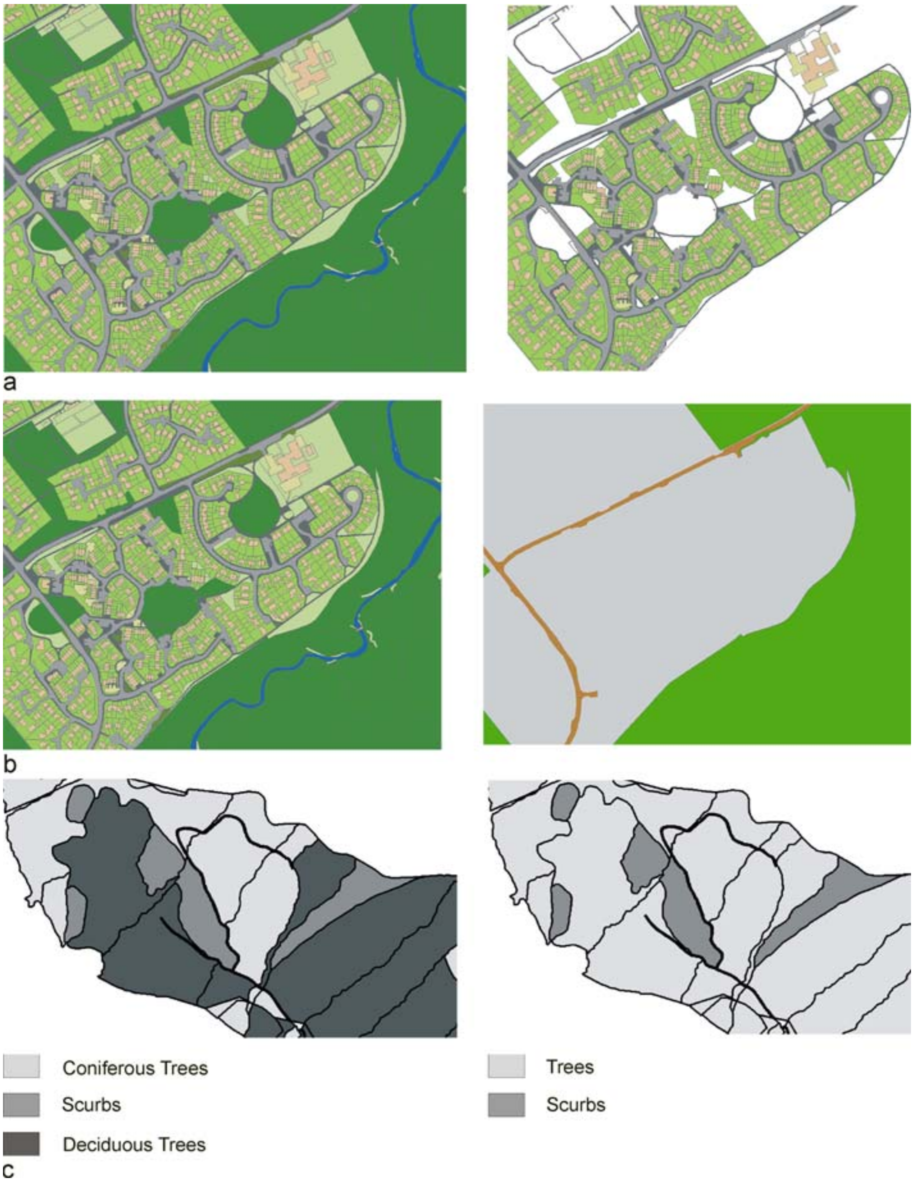
Generalization and Symbolization, Figure 3
DLM, DCM, Model and Cartographic Generalization

containing some abstraction of the real world, 2) a set of algorithms for aggregating objects in that database (model generalization), 3) a library of symbols with which to render the objects according to various themes, and 4) a set of algorithms focusing on improving the legibility of those symbolized objects (cartographic generalization). The database containing that first abstraction is typically called a digital landscape model (DLM – Fig. 3) [7]. The DLM might be created by digitizing paper maps, or from photogrammetric techniques applied to remotely sensed imagery. Typically a notional scale is associated with the DLM database though it is more apposite to talk of level of detail. Data from the database can be symbolized and visualized directly via cartographic techniques. Alternatively a database of lower semantic and geometric resolution can first be derived (via model generalization) – creating different digital cartographic models (DCM – Fig. 3) before cartographic generalization techniques are applied to produce different maps.

Altering the theme, and level of detail enables different phenomena and different properties to be portrayed. Sometimes the emphasis is on precision of location, or of shape (important in the map interpretation process). In other circumstances, the emphasis may be on connectivity at the expense of other properties and qualities. Maps of transportation networks (such as the London Underground) are a nice example of the need to emphasize connectivity over geographical location. Irrespective of theme, in all cases a map (digital or paper) reflects a compromise in design – a compromise between wanting to convey information unambiguously but not having enough room (given the minimum size of symbology) to show all that information. In this sense the process of design is about making sense of things – the cartographer perhaps working from a mental thumbnail sketch by which their solution reflects the requirements of the user in terms of their needs, which in turn governs and constrains the representation of each feature in the map.

Various methodologies have been proposed that try to capture this design process within an automated environment. Considerable research effort has gone into creating algorithms that mimic these human techniques. These techniques are not applied in isolation, but rather in concert, and in varying degree, across the map, depending on the density of information, and the type of phenomenon being mapped, and of course, the theme and scale. Therefore in addition to algorithms that mimic these techniques, a framework is required that can orchestrate this whole design process, together with some evaluation methodologies required to assess the quality of the solution produced within such a framework. Next is a review of generalization techniques under the headings of model and cartographic generalization.

Model Generalization The objective of model generalization techniques is to reclassify and reduce down the detail, thus giving emphasis to entities associated with the broader landscapes – enabling us to convey the extent of the forests rather than see the trees, or to see the island chain along the plate margin, rather than the individual island. The model generalization process is not concerned with issues of legibility and visualization. It is more useful

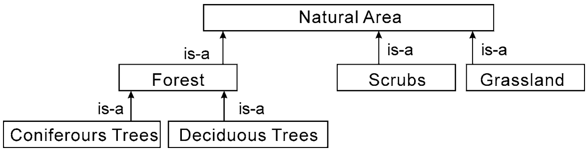


Generalization and Symbolization, Figure 4 **a** Selection, **b** Aggregation and **c** Classification. (Mapping is Ordnance Survey ©Crown Copyright. All rights reserved)

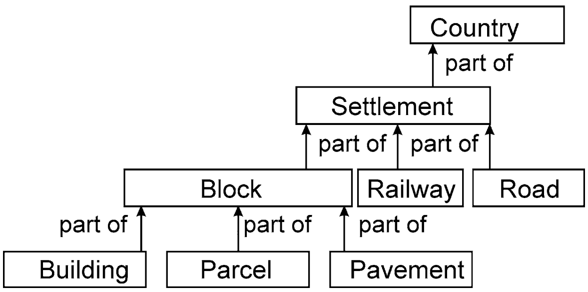
to view it as a filtering process; a set of techniques concerned with 1) selection of phenomena according to theme, and 2) the classification and aggregation of phenomena. As the name suggests, selection is the (straightforward) process of selecting a subset of all classes of objects falling within a specified region (Fig. 4). The selection process is governed by task, which in turn tends to define both the intended scale and theme. The long tradition of topographic and thematic mapping often acts as a basis for specifying content, and thus which classes of objects are selected.

Typically model generalization precedes cartographic generalization. It may also be required in response to a non-visual query, or as a prerequisite to data analysis. For example the question 'what modes of travel exist between the cities of Edinburgh and Glasgow?' requires us to aggregate together phenomena at the fine scale (in this case dense regions of buildings) in order to define the extent and general location of these two entities. Only then can the major roads connecting these two urban centers be identified.

Composite or higher order objects are formed via the process of thematic and spatial abstraction. In thematic abstraction the number of distinct attributes of objects in the database is reduced. In spatial abstraction the number objects is reduced by means of aggregation or elimination. Thematic abstraction often triggers spatial abstraction. For instance objects having similar attribute structure can be categorized into classes under the process of classification.



Generalization and Symbolization, Figure 5 Example of a taxonomy



Generalization and Symbolization, Figure 6 Example of a partonomy

Each object then becomes an instance of a particular class and that class defines an object's properties in terms of its attribute structure. If different classes share some attributes then a super class or parent class can be created whose attributes are the common attributes of its child classes. This creates a hierarchy where complex classes are present at the detailed (low end of a hierarchy) and increasingly abstracted classes are present as one travels up the hierarchy. This type of hierarchy is called a taxonomy or classification hierarchy (Fig. 5) and can be used as a basis for classification of data ('classification' Fig. 4).

Another complimentary hierarchy useful in the creation of composite objects is a partonomy. Whereas a taxonomy refers to a 'is-a' relationship, a partonomy refers to 'part-of' relationships between parent and child classes – reflecting more of a functional and conceptual division of geographic space (Fig. 6) [8]. Over large changes in scale it is necessary to aggregate objects belonging to different classes in order to create composite objects. A prototypical view of a city might be defined as a dense collection of municipal and industrial buildings, and multi modal transportation infrastructures. Once represented in partonomic form, it can be used as a basis for combining such objects together ('aggregation' Fig. 4).

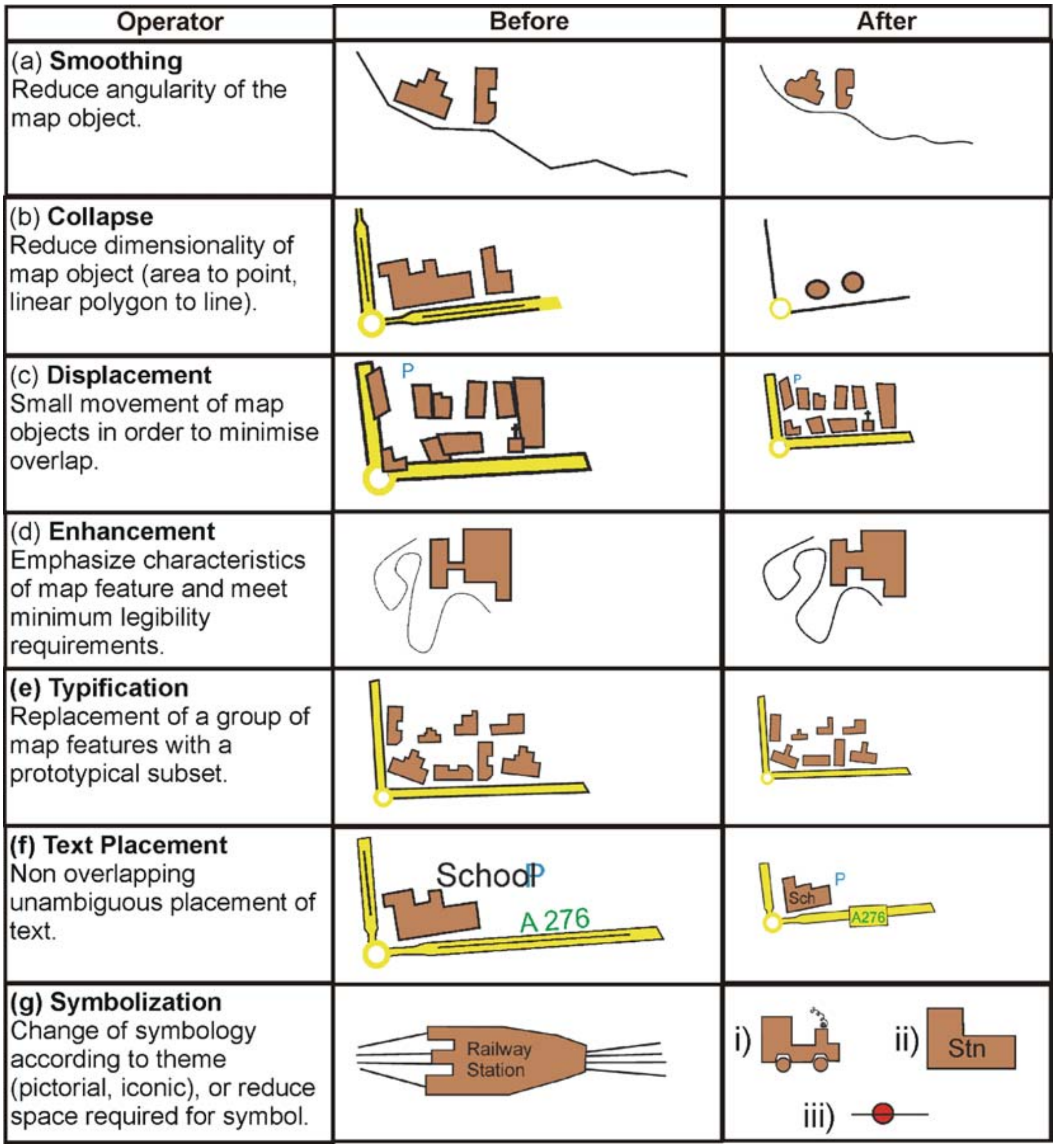
In addition to the techniques of selection and aggregation, there is 'simplification' – which is defined as the process of reducing the number of geometric points used to store the physical location or extent of a geographic object. One can envisage many points being used to record the detail of the outline of a gothic cathedral, or the sinuous path of a low lying river. The challenge of simplification is to reduce the number of points used to store the representation of such features, but in a way that still conveys their essential shape and location. Successful simplification reduces storage requirements and processing time. Once the model generalization process is completed, the challenge is then to render those objects into some map space (whether it is for paper production, or as part of a digital interactive environment – in either a desktop or mobile environment).

Cartographic Generalization Cartographic generalization involves symbolizing the selected data, and applying a set of techniques that optimally convey the salient characteristics of that data, including careful placement of associated text. Symbols used to represent spatial objects from the source database need to be visible to the naked eye. As the scale reduces the amount of space available decreases thus creating competition for space among the symbology. To retain clarity and to represent the information effectively a range of techniques are applied such as symbolization, smoothing, simplification, grouping, enhancement, displacement, and text placement (Fig. 7). These techniques (often applied in combination), seek to give prominence to the essential qualities of the feature portrayed (that rivers retain their sinuous and connected form, and buildings retain their anthropogenic qualities – such as their angular form). Different combinations, amounts of application, and different orderings of these techniques can produce different yet aesthetically acceptable solutions. The focus is not on making changes to information contained in the database, but to solely focus upon avoiding ambiguity in the interpretation of the image. The process is one of compromise reflecting the long held view among cartographers that making maps involves telling small lies in order to tell the truth!

Analysis, Synthesis and Evaluation of Cartographic Solutions

For any given cartographic conflict, one can envisage a number of viable solutions. The choice of solutions will depend on: the density of features, their position relative to one another, and their importance relative to the intended theme. Trying to create alternate viable solutions (synthesis), and then choosing a solution amongst that choice requires two things: 1) an initial analysis phase in which the conflicts are identified (analysis) and a form of evaluation such that the quality of the solution can be assessed (evaluation). Failure to find an adequate solution might either result in further analysis of the conflict or flagging unresolved conflicts and drawing these to the attention of the user.

The analysis phase is akin to the eyes of the cartographer and involves making assessment of the degree of severity

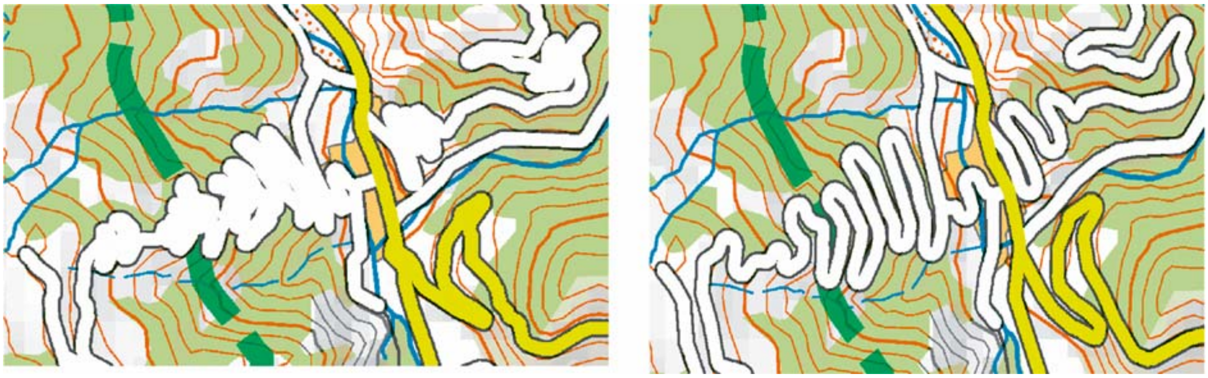


Generalization and Symbolization, Figure 7 Cartographic generalization operations

of the conflict (extent and complexity and composition). A broad and extensive set of cartometric techniques have been developed to measure the various qualities inherent among a set of map objects. This analysis is required because the goal is to ensure minimum disruption in those qualities during the cartographic generalization process. Many shape and pattern metric techniques have been proposed to measure and minimize the effects of cartographic generalization [9,10]. These are often applied in the analysis phase, and again in the evaluation phase. The best solution among a set of candidate solutions might be the one that has resolved the conflict (improved its legibility), whilst producing the least amount of change among the various cartometric measures (in terms of topology, orientation, area, shape and distance).

Modeling the Generalization Process

The selection and application of generalization techniques, the creation of candidate solutions and their evaluation requires some framework in which this can all take place. Because of the interdependent nature of geographic phenomena, it is rare that changes can be made without having to consider the broader context. For example the solution in Fig. 7c is only appropriate because there is sufficient space for the objects to be displaced into. If buildings have



Generalization and Symbolization, Figure 8 Example output from the IGN's agent based system

to be aggregated in one part of the map (perhaps because of the density of features) then for reasons of consistency, this needs to be applied in other similar instances. Procedural and heuristic knowledge needs to be incorporated within these frameworks so that the solutions most likely to be successful can be applied first. Among the various 'frameworks' explored, two are worthy of mention: rule based approaches, and constraint based approaches.

Since the cartographic design process involves decision making and heuristics ('rules of thumb'), it was assumed that knowledge based approaches (expert systems) could be used to model the process – using a rule based approach. These systems used either a predetermined rule execution sequence or an inference engine to control the execution sequence in applying various techniques. They consisted of three main parts: a knowledge base, an inference engine and a user interface. The knowledge base contained a set of rules, facts or procedures. The inference engine controlled the generalization process by making use of the rules and procedures in the knowledge base. The user interface supported the process of data selection and a mechanism for adding or updating rules in the knowledge base [11].

More recently generalization research has focused on an holistic view of the process acknowledging the knock on effects of generalization and the interdependent nature of the solution. Currently there is much interest (and promise) in using constraint based approaches – where the aim is to find a state whereby the maximum number of constraints can be satisfied. In this context, much research effort has been devoted to agent based methodologies – in which each object in the database is modeled as an agent – an object oriented concept in which the object has goals, behaviors, and a capacity to communicate with other agents. These are referred to as 'multi agent systems'. The goals reflect those of the generalization process – namely to efficiently render the object without ambiguity. The agent makes decisions about its representation based on its own goals whilst considering the goals and constraints of its neighbors. Ideas have included a hierarchy of agents in which higher order agents are concerned with broader contexts and distribution of agent classes, whilst agents at the individual object level are concerned with the specific representation of individual objects. The AGENT [12] project is one project which has been developed into a commercial system that now supports a number of national mapping agencies, notably the National Mapping Agency of France (IGN). Figure 8 shows the result from the Carto2001 project [13].

By partially incorporating the decision making process within both rule based and agent based systems, the balance of decision making has shifted away from the human to the machine. This has presented some real challenges in the design of interfaces that are intuitive to use, allowing the user to specify their mapping requirements in a simple and efficient manner within a very complex system. Researchers have challenged the idea of totally autonomous solutions, arguing that interaction is critical to ensuring that the user remains very much part of the design process. The idea of semi autonomous generalization techniques, involving the user in critical evaluation tasks reflects a more collaborative approach to design. Coupled with machine learning techniques, this scenario might enable capture of design heuristics – thus gradually improving the sophistication of proffered solutions.

Key Applications

The idea that map generalization is some 'cartographic end process' belies its importance in supporting five key activities:

Cartographic Assistant

The existence of many different generalization techniques means that a 'cartographic toolbox' is available for use by a trained cartographer. Research efforts have yielded a set of algorithms able to analyze map content, and to consistently generalize classes of objects in clearly defined ways. In this collaborative environment, such systems have the capacity to improve the quality of cartographic training, ensure quality control in the design process and enable refinement in the adjustment of parameters used to control generalization techniques.

Map Generalization Service

In the absence of the cartographer, and in the context of GIS, users (with limited cartographic knowledge) require assistance in the rapid design and delivery of cartographic products – often via the Internet, that can vary in theme and scale according to task. Completely autonomous solutions (with no user intervention) have proved to be very difficult to design, but in any case are not desirable where meaning is often derived through interaction and exploration of the data. The idea of a map generalization service is that maps can be delivered over the Internet in response to user requests – which in turn has led to a focus on the pre-processing of solutions, in which intermediate solutions are stored in a multiple representation database (MRDB).

Populating Multiple Representation Databases

There currently exist multiple, often disconnected 'silo' databases containing data at different levels of detail. The vision is that model generalization techniques are applied to data captured at the finest detail in order to create a hierarchical framework of increasingly aggregated geographic phenomena (from house, to suburb, to city to region, to country) – in effect a semantically indexed structure from which different scale linked phenomena can be extracted and queried. The benefit of this approach is consistency and 'lineage' (provenance) by which the source objects from which the higher order geographies have been created can be identified. This can support both data integration, and hugely facilitate the data update process. The existence of MRDB can also support on-the-fly generalization and instantaneous delivery of geographic data over the Internet and mobile devices [14,15].

Spatial Data Integration Service

Considerable 'value add' comes from the sharing and integration of data. Integration of geographic data is beset by a host of challenges receiving considerable attention – notably in development of shared data schemas, and addressing ontological issues linked to culture, original purpose and conceptual understandings of place. Many of these issues relate to the notional scale at which the data was originally captured. Model generalization techniques can play a critical role in aggregating data according to shared partonomic and taxonomic classification methodologies.

Future Directions

Generalization in the context of geographical information science has an importance beyond traditional cartographic lines. It has everything to do with revealing and giving emphasis to properties inherent among geographic phenomena – and therefore has important cross over with ideas of design (making sense of things), data mining and geo-visualization [16]. The aggregation of phenomena is dependent on taxonomic and partonomic hierarchies, which themselves reflect complex functional and contextual interdependencies inherent among geographic phenomena. In this sense, issues of generalization are central to meaningful interrogation and analysis of all geographic information.

Cross References

► Conflation of Geospatial Data
► Data Analysis, Spatial
► Data Warehouses and GIS
► Exploratory Visualization
► Generalization, On-the-Fly
► Hierarchical Spatial Models
► Hierarchies and Level of Detail
► Map Generalization
► Mobile Usage and Adaptive Visualization
► Scale, Effects
► Web Mapping and Web Cartography

Recommended Reading

1. Taylor, P.J.: Is there a Europe of cities? World cities and the limitations of Geographical Scale Analyses. In: Sheppard, E., McMaster, B. (eds.) Scale and Geographic Inquiry: Nature, Society, and Method, pp. 213–235 Blackwell Publishing, Malden, MA (2004)
2. Leitner, H.: The Politics of Scale and Networks of Spatial Connectivity: Transnational Interurban networks and the Rescaling of Political Governance in Europe. In: Sheppard, E., McMaster, B. (eds.) Scale and Geographic Inquiry: Nature Society and Method, pp. 236–255 Blackwell Publishing, Malden, MA (2004)
3. Hartshorne, R.: The Nature of Geography: A Critical Survey of current thought in the light of the past. PA: Association of American Geographers, Lancaster (1939)
4. Dykes, J.: Exploring Spatial Data Representation with Dynamic Graphics, Computers and Geosciences. **23** 345–370 (1996)
5. Fisher, P., Dykes, J., Wood, J.: Map Design and Visualization. The Cartographic Journal. **30** 136–142 (1993)

6. Monmonier, M.: How to Lie with Maps. University of Chicago Press, Chicago (1991)
7. Brassel, K.E., Weibel, R.: A review and conceptual framework of automated map generalization. International Journal of Geographical Information Systems. **2** 229–244 (1988)
8. Molenaar, M.: An Introduction to the Theory of Spatial Object Modeling for GIS. Taylor and Francis, London (1998)
9. Peng, W., Sijmons, K., Brown, A.: Voronoi Diagram and Delaunay Triangulation Supporting Automated Generalization. Paper presented at 17th ICA/ACI, Barcelona, Spain, 1995
10. Mackaness, W.A., Beard, M.K.: Use of Graph Theory to Support Map Generalization. Cartography and Geographic Information Systems. **20** 210–221 (1993)
11. Buttenfield, B.P., McMaster, R.B.: Map Generalization: Making Rules for Knowledge Representation. Longman, London (1991)
12. Lamy, S., Ruas, A., Demazeau, Y., Jackson, M., Mackaness, W.A., Weibel, R.: The Application of Agents in Automated Map Generalization. Paper presented at Proceedings of the 19th International Cartographic Conference, Ottawa 1999
13. Lecordix, F., Lemarie, C.: Managing Generalization Updates in IGN Map Production. In: Mackaness, W.A., Ruas, A., Sarjakoski, L.T. (eds.) Generalization of Geographic Information: Cartographic Modeling and Applications. Elsevier, Oxford (2007)
14. Elias, B., Brenner, C.: Automatic generation and application of landmarks in navigation data sets. Paper presented at Developments in Spatial Data Handling, Proceedings of the 11th International Symposium on Spatial Data Handling, Leicester, UK, 2004
15. Dias, E., Beinat, E., Scholten, H.: Effects of Mobile Information Sharing in Natural Parks. Paper presented at ENVIROINFO 2004, 18th International Conference Informatics for Environmental Protection, CERN, Geneva, Switzerland, 2004
16. Dykes, J., MacEachren, A.M., Kraak, M.-J.: Exploring Geovisualization. Elsevier, London (2004)
17. Barrault, M., Regnauld, N., Duchene, C., Haire, K., Baeijs, C., Demazeau, Y., Hardy, P., Mackaness, W., Ruas, A., Weibel, R.: Integrating multi agent, object oriented and algorithmic techniques for improved automated map generalization. In: Proceedings of the 20th International Cartographic Conference, **3**, 2110–2116, Beijing, China (2001)
18. Devogele, T., Trevisan, J., Ranal, L.: Building a Multi Scale Database with Scale Transition Relationships. In: Kraak, M.-J., Molenaar, M., Fendel, E.M. (eds.) Advances in GIS Research II, Proceedings of the 7th International Symposium on Spatial Data Handling, pp. 337–352. Taylor and Francis, Delft, London, (1996)
19. João, E.M.: Causes and Consequences of Map Generalization. Taylor and Francis Ltd., London (1998)
20. Kilpelainen, T., Sajakoski, T.: Incremental generalization for multiple representations of geographical objects. In: Muller, J.C., Lagrange, J.P., Weibel, R. (eds.) GIS and Generalization: Methodology and Practice, pp. 209–218. Taylor & Francis, London (1995)
21. Mackaness, W.A., Ruas, A., Sarjakoski, L.T.: Generalization of Geographic Information: Cartographic Modeling and Applications. Elsevier, Oxford (2007)
22. McMaster, B.R., Shea, K.S.: Generalization in Digital Cartography. Association of American Geographers, Washington, D.C. (1992)
23. Müller, J.C., Lagrange, J.P., Weibel, R.: GIS and Generalization: Methodology and Practice. In: Masser, I., Salge, F. (eds.) GISDATA 1. Taylor and Francis, London (1995)
24. Sheppard, E., McMaster, R.B.: Scale and Geographic Inquiry: Nature Society and Method. Blackwell Publishing, Malden, MA (2004)
25. Weibel, R., Dutton, G.: Generalizing Spatial Data and Dealing with Multiple Representations. In: Longley, P., Goodchild, M.F., Maguire, D.J., Rhind, D.W. (eds.) Geographical Information Systems, pp. 125–156. John Wiley, New York (1999)

Generalization, On-the-Fly

ROBERT WEIBEL, DIRK BURGHARDT
Department of Geography, University of Zurich, Zurich, Switzerland

G

Synonyms

Real-time generalization; Dynamic generalization; Online generalization; Hierarchical data structures

Definition

Map generalization defines the process of producing maps at coarser levels of detail (LOD), while retaining essential characteristics of the underlying geographic information. On-the-fly generalization, then, denotes the use of automated generalization techniques in real time. According to [1], this process creates a temporary, generalized dataset exclusively for visualization, not for storage or other purposes. On-the-fly generalization is intimately linked to highly interactive applications of cartography such as web mapping, mobile mapping [e. g., in location-based services (LBS)], and real-time decision support systems (e. g., in disaster and evacuation management) that involve multiple spatial scales. As it takes place in a highly interactive setting, the cartographic quality requirements are typically relaxed compared to traditional, high-quality paper maps. On the other hand, (near) real-time behavior is imperative.

Solutions that satisfy the above requirements can generally be assigned to two groups. The first group of approaches relies on fast map generalization algorithms that generate coarser levels of detail in real time. The second group utilizes hierarchical spatial data structures. In both cases, the generalization operations that are implemented are generally rather simple from a functional point of view, compared to the cartographically more sophisticated, yet computationally more costly algorithms that are typically used in the production of high-quality paper maps. Closely related to on-the-fly map generalization is progressive vector data transmission (i. e., the transmission, over a network, of vector datasets at progressively finer detail).

Historical Background

For centuries, cartography was exclusively devoted to the production of static paper maps. Even with the introduction of the computer to cartography in the 1960s and the growing use of interactive computer systems in the 1980s the situation did not change much. Paper was still the main output medium, and screen maps were commonly only used for editing and proofing in paper map production, rather than as end products. Consequently, research in automated map generalization—despite the fact that it dates back to the early days of computer cartography and geographical information systems (GIS)—focused primarily on achieving high cartographic quality in the generalization process, while largely neglecting computational efficiency. While this preference for graphic quality over efficiency may sound odd to the computer scientist, it did make sense from a cartographic perspective, bearing in mind that firstly, map generalization is an ill-defined process and nontrivial to automate [2], and secondly, since the end products were static, improved cartographic quality at the expense of added computing time could easily be tolerated.

The advent of interactive personal computers in the 1980s and, more importantly, of the world wide web (WWW) in the early 1990s brought new requirements for cartography and map generalization. Since the usage and interaction with web mapping services are highly time-critical, scale-changing has to take place in real time, essentially demanding on-the-fly generalization. However, despite early solutions for on-the-fly line simplification and object selection using reactive data structures [3,4], researchers in automated map generalization continued to place little emphasis on on-the-fly generalization throughout the 1990s. Operational web map services such as mapquest.com, mapblast.com, or map24.com rely on pragmatic solutions, involving offline production of multirepresentation databases containing multiple LOD, restricting the on-the-fly part to the real-time retrieval and display of the LOD that matches the given zoom level as well as real-time placement of map labels and symbols (e. g., icons for points-of-interest).

In recent years, however, repeated calls have been made for a redefinition of cartography and map generalization [5,6]. New forms of mapping applications such as mobile mapping in LBS or real-time decision support systems go beyond web mapping as it's best known, in the form of the services mentioned above. In addition to the requirement of real-time map delivery, these new mapping applications demand adaptation and personalization of the thematic map content to the given user query and context. Thus, precomputing and storing potential visualizations as in "classical" web mapping is no longer a feasible solution; true on-the-fly generalization capabilities are needed. Researchers have started to respond to these new challenges in recent years.

Scientific Fundamentals

On-the-Fly Generalization Versus Multirepresentation Databases

As mentioned in the preceding section, true on-the-fly generalization is not to be equated with the mere retrieval and display of pregeneralized LOD from a multirepresentation database (MRDB). Hence, there is no need to dwell on MRDB further. However, it should be emphasized that MRDBs remain an active and important research area [7,8]. For instance, many public and private mapping organizations have large holdings of digitized maps at different scales and thus are interested in linking these together so that updates can be propagated automatically from detailed to less detailed representations, allowing incremental updates [9].

Characteristics

The main characteristics of on-the-fly generalization are (see also [10]):

- A temporary, reduced scale (generalized) dataset/map is generated for visualization purposes from a spatial database.
- The map has to meet the user preferences (e. g., personalized content) and the technical display specifications (i. e., typically low screen resolution and small screen size).
- The scale of the resulting map may vary (particularly due to zooming operations) and is not predefined.
- The generalization process must be accomplished automatically and no user interaction is possible, e. g., to check the result before publishing.
- The resulting map must appear on the display within a few seconds, as the user does not want to wait.
- On the web and on mobile devices, there is an additional problem of limited network bandwidth.

Techniques for on-the-fly generalization follow two main tracks, either making use of efficient algorithms that allow generation of coarser LOD in real time, or exploiting hierarchical spatial data structures.

On-the-Fly Generalization by Algorithms

Since on-the-fly generalization is a time-critical task, generalization algorithms that are suited for this purpose must be fast and/or they must be supported by precomputed data structures or attributes. In principle, all known generalization algorithms that run in linear or logarithmic time make candidates for on-the-fly generalization. One example of such fast algorithms is simple selection algorithms that merely rely on precomputed attributes, such as the Horton stream ordering scheme used for river network selection [10]. Attribute-based selection is also straightforward to implement, as exemplified by the system described in [11] that uses the extensible stylesheet language transformation (XSLT) mechanism to generate real-time, generalized maps. An extended version of this system [12] offers a range of well-known algorithms: feature selection by object class, area selection by minimum/maximum value, line selection by minimum/maximum length, contour line selection by interval, line simplification by the Douglas–Peucker algorithm, line simplification by the Lang algorithm, line smoothing by Gaussian filtering, and building outline simplification. Another system for on-the-fly generalization that makes use of a combination of simple (and efficient) algorithms is described in [13].

The algorithms discussed so far have in common that they were originally not specifically developed for on-the-fly generalization of potentially large datasets. They are merely useful for this purpose because they are so simple that they require relatively little computational effort. An algorithm that specifically targets dynamic generalization is presented in [14]. This algorithm performs line simplification of large map datasets through a novel use of graphics hardware (frame buffer, color buffer, stencil buffer, depth buffer) using a hybrid vector/raster-based approach. For interactive visualization, presimplified maps of different LOD are organized in a hierarchical data structure, the Data Tree. the solution presented in [14] thus represents a hybrid between algorithmic approaches to on-the-fly generalization and those methods that fully rely on hierarchical data structures (to be discussed next section).

The above examples implement only algorithms that are restricted to rather simple generalization operations without consideration of their spatial context, such as selection, line simplification, line smoothing and polygon aggregation (e. g., by the convex hull). More complex, contextual generalization operations such as feature displacement or typification—necessary to achieve high cartographic quality—commonly require iterative optimization techniques that are generally not suited to real-time applications (for an overview, see [15]). A possible solution to speed up displacement computation consists in using interpolation, or “morphing”, between the geometries of two LOD [16]. This approach, however, requires at least two LOD whose vertices of corresponding map objects have been correctly matched and linked in an MRDB. A more realistic approach to achieving more complex generalization behavior that is nevertheless efficient is by using auxiliary data structures, as discussed in the next section.

On-the-Fly Generalization by Hierarchical Data Structures

Map generalization results in hierarchies of progressively coarser maps. Thus, it is only natural that hierarchical spatial data structures are exploited in map generalization, and even more prominently in speeding up on-the-fly map generalization. This section discusses selected examples of solutions that rely exclusively on the hierarchical representation of spatial data in tree data structures. These examples have in common that they try to establish variable scale data structures, thus avoiding the redundant data storage typical of multiple representations using a stack of LOD.

The earliest proposal of a tree data structure for on-the-fly generalization was already mentioned in the historical overview: the Binary Line Generalization (BLG) Tree [3]. It uses the classic of line simplification, the Douglas–Peucker algorithm, to precompute the order of elimination of the vertices of a line. The vertex numbers and associated tolerance values are then stored in a binary tree. At run time, the tree can be descended down to the level that matches the resolution of the target map and the corresponding vertices retrieved for rendering. As the BLG tree is restricted to organizing single line objects, it cannot be used for the spatial organization (e. g., indexing) of multiple map objects. This restriction is overcome by the Reactive Tree [4], an extension to the R-tree [17] that stores importance levels for map objects (with important objects stored higher in the tree). The Reactive Tree is dynamic, allowing inserts and deletes.

The BLG and Reactive Tree data structures are not suited to the generalization of polygonal maps [18], as they do not represent the special nature of an area partitioning of adjacent polygons. This deficiency led to the development of the Generalized Area Partitioning (GAP) Tree which defines successive hierarchies of aggregations of adjacent polygons in a polygonal map. A system which uses the BLG Tree (for line simplification), the reactive Tree (for map object selection), and the GAP Tree (for area aggregation) together is reported in [1], containing also a description of the GAP Tree data structure. Recently, a new, topological version of the GAP Tree was introduced [18] which combines the use of the BLG Tree and the Reactive Tree and avoids redundant storage and sliver polygons along the boundary of neighboring polygons, problems associated with the original GAP Tree.

The use of hierarchical data structures for on-the-fly generalization of point distributions commonly found in thematic maps (e. g., animal observation data, distributions of disease occurrences) and LBS (e. g., points-of-interest) is reported in [19,20]. Two methods are proposed. The first one uses a quadtree to index the original points to successively coarser, aggregated levels. At run time the original points are then replaced by the centroids of the quadtree cells corresponding to the appropriate resolution (i. e., scale of the target map). The disadvantage of this first solution is that the output point pattern will be aligned to the (regular) quadtree pattern, creating an unnatural arrangement. Hence, a second proposed solution uses a hierarchical tessellation of the map space that corresponds to the semantics of the data points. In the example shown in [19,20], animal observation data are mapped to the network hierarchy of drainage basins, as these are bounded by ridges that often also form physical barriers to animal movement.

Related Issues

In recent years, research interest has started to develop into methods for the progressive transmission of vector map data. This interest is motivated by the very same reason that prompted the earlier development of progressive techniques for the transmission of raster images over the WWW, implemented today in standard image formats: the need to access large datasets in distributed, bandwidth-limited computing environments. Progressive vector data transmission shares with on-the-fly generalization the aim to represent map data at successively coarser or finer LOD, respectively. While the aim of progressive vector data transmission is to ultimately transmit the *entire* dataset, the user will initially only receive a coarse representation of the map data, followed by progressive refinements, until the full map has been transmitted. Any of the intermediate refinement steps represents a generalization of the full map. Hence, there is also a strong similarity (or even congruence) of methods between progressive vector data transmission and on-the-fly generalization. In comparison to progressive methods for image data, equivalent methods for vector data are inherently more complex to achieve and hence still very much in the research stage. Starting from initial conceptual work [21], solutions have been proposed for the "continuous generalization" of buildings for LBS [22], for an MRDB architecture in the context of LBS applications [23], and for line and polygon data [24].

Label and icon placement on screen maps is a further issue that is closely related to online-generalization, for two reasons. First, the selection of map labels and/or map icons is driven by the same principles—scale, semantics, available map space—as the selection of other map objects. Second, the placement of map labels and map icons shares many similarities with displacement operations in map generalization. While many algorithms exist for offline placement of map labels and icons, real-time labeling (e. g., for mobile maps in LBS) has only rarely been addressed in the literature so far [25].

Finally, web generalization services should be mentioned. On-the-fly generalization is typically linked to web and/or mobile mapping applications, hence to applications that take place in distributed computing environments and client/server architectures. Therefore, the recently initiated move toward the exploitation of service-based architectures in map generalization [26,27] nicely corresponds to the evolution of on-the-fly generalization.

Key Applications

As has become obvious from the preceding discussion, on-the-fly map generalization is still very much a research area. The development of appropriate techniques targets a variety of applications which have in common that they are highly interactive and have requirements for (near) real-time visualization with adaptable scale and content. Following are a few examples of such applications.

Web Mapping

As mentioned in the Historical Background section the evolution of web mapping has provided the initial setting that prompted the need for on-the-fly generalization capabilities. For many years, web mapping largely defined the requirements for on-the-fly generalization. Today, however, it has been superseded as a trendsetter by less mainstream applications.

Adaptive Zooming

Adaptive zooming is a capability that is still sorely lacking in many interactive mapping systems. It denotes "the adjustment of a map, its contents and the symbolization to target scale in consequence of a zooming operation" [28]. As follows from this definition, adaptive zooming also requires some sort of on-the-fly generalization. A pragmatic solution that uses on-the-fly algorithms for the simple generalization operations in combination with LOD as substitutes for complex generalization operations is presented in [28].

Mobile Cartography and LBS

LBS have given a new direction to cartography and GIS. They place the user in the center of the map; the map display needs to adapt to the user's changing location; and the map display needs to be adapted (or personalized) to the user's specific information requirements. Furthermore, mobile devices are bound to impose more stringent technical limitations than commonly encountered in cartography (e. g., low resolution and small size of the display screen, low bandwidth, unreliable network connectivity). An overview discussion of the requirements and research perspectives of LBS, including the need for on-the-fly generalization, can be found in [29].

Real-Time Decision Support Systems

GIS are used a great deal as tools for decision support. While most uses of spatial decision support systems (SDSS) do not have real-time requirements, new applications have recently started to appear that do involve decision making in response to real-time data feeds. Examples include emergency service dispatching, evacuation route planning, and disaster management [30].

Future Directions

As [18] notes, data structures supporting variable scale datasets—and hence also solutions for on-the-fly map generalization—are still very rare. On the other hand, there are a growing number of applications that require functionality for real-time adaptation of spatial datasets and maps to the scale and purpose of the target display. Hence, it can be expected that increasingly more sophisticated solutions will complement or supersede the rather pragmatic, usually LOD-based techniques commonly used today. In addition to the development of new techniques, there is also room for improvement by *combining* existing methods. First, individual real-time algorithms may be combined to create more comprehensive solutions, as exemplified by [18]. In the future, this approach may also benefit from the current trend towards web-based architectures [26]. A second track may exploit the potential of combining MRDB- and LOD-based techniques and on-the-fly generalization, illustrated by the (still rather pragmatic) solution presented in [28].

Cross References

► Indoor Positioning

Recommended Reading

1. van Oosterom, P., Schenkelaars, V.: The development of an interactive multiscale GIS. Int. J. Geogr. Inf. Syst. **9**, 489–507 (1995)
2. Weibel, R., Dutton, G.: Generalizing spatial data and dealing with multiple representations. In: Longley, P.A., Goodchild, M.F., Maguire, D.J.,Rhind, D.W. (eds.) Geographical Information Systems: Principles, Techniques, Management and Applications, 2nd edn., pp. 125–155. Wiley, Chichester (1999)
3. van Oosterom, P., van den Bos, J.: An object-oriented approach to the design of geographic information systems. Comput. Graphics **13**, 409–418 (1989)
4. van Oosterom, P.: A storage structure for a multiscale database: the reactive-tree. Comput., Environ. Urban Syst. **16**, 239–47 (1992)
5. Jones, C.B., Ware, J.M.: Map generalization in the web age. Int. J. Geogr. Inf. Sci. **19**, 859–870 (2005)
6. Mackaness, W.A.: Automated cartography in a bush of ghosts. Cartogr. Geogr. Inf. Sci. **33**, 245–256 (2006)
7. Hampe, M., Sester, M.: Generating and using a multi-representation database (MRDB) for mobile applications. Paper presented at the ICA Workshop on Generalization and Multiple Representation, Leicester, 20–21 August 2004. Available via: http://ica.ign.fr/Leicester/paper/hampe-v2-ICAWorkshop.pdf. Accessed 12 Feb 2007
8. Viaña, R., Magillo, P., Puppo, E., P.A. Ramos: Multi-VMap: a multi-scale model for vector maps. GeoInformatica, **10**, 359–394 (2006)
9. Anders, K.-H., Bobrich, J.: MRDB Approach for Automatic Incremental Update. Paper presented at the ICA Workshop on Generalization and Multiple Representation, Leicester, 20–21 August 2004. Available via: http://ica.ign.fr/Leicester/paper/Anders-v1-ICAWorkshop.pdf. Accessed 12 Feb 2007
10. Rusak Masur, E., Castner, H.W.: Horton's ordering scheme and the generalisation of river networks. Cartogr. J. **27**, 104–112 (1992)
11. Lehto, L., Sarjakoski, L.T.: Real-time generalization of XML-encoded spatial data for the web and mobile devices. Int. J. Geogr. Inf. Sci. **19**, 957–973 (2005)
12. Sarjakoski, L.T., Sarjakoski, T.: A use case based mobile GI service with embedded map generalisation. Paper presented at the ICA Workshop on Generalization and Multiple Representation, Leicester, 20–21 August 2004. Available via: http://ica.ign.fr/Leicester/paper/Sarjakoski-v2-ICAWorkshop.pdf. Accessed 12 Feb 2007
13. Glover, E., Mackaness, W.A.: Dynamic generalisation from a single detailed database to support web based interaction. In: Proceedings 19th International Cartographic Conference, Ottawa, 14–21 August (1999)
14. Mustafa, N., Krishnan, S., Varadhan, G., Venkatasubramanian, S.: Dynamic simplification and visualization of large maps. Int. J. Geogr. Inf. Sci. **20**, 273–302 (2006)
15. Mackaness, W.A., Ruas, A., Sarjakoski, L.T. (eds.): Generalisation of Geographic Information: Cartographic Modelling and Applications. Elsevier, Amsterdam, Boston, Heidelberg, London, New York, Oxford, Paris, San Diego, San Francisco, Singapore, Sydney, Tokyo (2007)
16. Sederberg, T.W., Greenwood, E.: A physically based approach to 2-D shape blending. Comput. Graphics **26**, 25–34 (1992)
17. Guttmann, A.: R-Trees: A dynamic index structure for spatial searching. ACM SIGMOD 13, 47–57 (1984)
18. van Oosterom, P.: Variable-scale topological data structures suitable for progressive data transfer: The GAP-face tree and GAP-edge forest. Cartogr. Geogr. Inf. Sci. **32**, 331–346 (2005)
19. Burghardt, D., Purves, R. S., Edwardes, A.J.: Techniques for on-the-fly generalisation of thematic point data using hierarchical data structures. In: Proceedings of the GIS Research UK 12th Annual Conference, Norwich, 28–30 April (2004)
20. Edwardes, A., Burghardt, D., Dias, E., Purves, R.S., Weibel, R.: Geo-enabling spatially relevant data for mobile information use and visualisation. In: Proceedings of the 5th International Workshop on Web and Wireless GIS, W2GIS 2005. Lecture Notes in Computer Science, vol. 3833, pp. 78–92. Springer-Verlag, Berlin, Heidelberg (2005)

G

21. Bertolotto, M., Egenhofer, M.J.: Progressive transmission of vector map data over the world wide web. GeoInformatica, **5**, 345–373 (2001)
22. Sester, M., Brenner, C.: Continuous generalization for visualization on small mobile devices. In: Peter Fisher (ed.) Developments in Spatial Data Handling–11th International Symposium on Spatial Data Handling, pp. 469–480. Springer Verlag, Berlin, Heidelberg, New York (2004)
23. Follin, J.-M., Bouju, A., Bertrand, F., Boursier, P.: Multiresolution extension for transmission of geodata in a mobile context. Comput. Geosci. **31**, 179–188 (2005)
24. Yang, B., Purves, R., Weibel, R.: Efficient transmission of vector data over the internet. Int. J. Geogr. Inf. Sci. **21**, 215–237 (2007)
25. Harrie, L, Stigmar, H., Koivula, T., Lehto, L.: An algorithm for icon labelling on a real-time map. In: Fisher, P. (ed.) Developments in Spatial Data Handling, pp. 493–507. Springer-Verlag, Berlin, Heidelberg, New York (2004)
26. Burghardt, D., Neun, M., Weibel, R.: Generalization services on the web—classification and an initial prototype implementation. Cartogr. Geogr. Inf. Sci. **32**, 257–268 (2005)
27. Foerster, T., Stoter, J.: Establishing an OGC web processing service for generalization processes. Paper presented at the ICA Workshop on Generalization and Multiple Representation, Portland, 25 June 2006. Available via: http://ica.ign.fr/Portland/paper/ICA2006-foerster_stoter.pdf Accessed 12 Feb 2007
28. Cecconi, A., Galanda, M.: Adaptive zooming in web cartography. Comput. Graphics Forum **21**, 787–799 (2002)
29. Jiang, B., Yao, X.: Location-based services and GIS in perspective. Comput., Environ. Urban Syst. **30**, 712–725 (2006)
30. Zerger, A., Smith, D.I.: Impediments to using GIS for real-time disaster decision support. Comput., Environ. Urban Syst. **27**, 123–141. (2003)

Generalized Minimum Spanning Tree

► Trip Planning Queries in Road Network Databases

Generalizing

► Hierarchies and Level of Detail

Genome Mapping

► Bioinformatics, Spatial Aspects

Geocollaboration

Shivanand Balram, Suzana Dragićević
Department of Geography, Simon Fraser University, Burnaby, BC, Canada

Synonyms

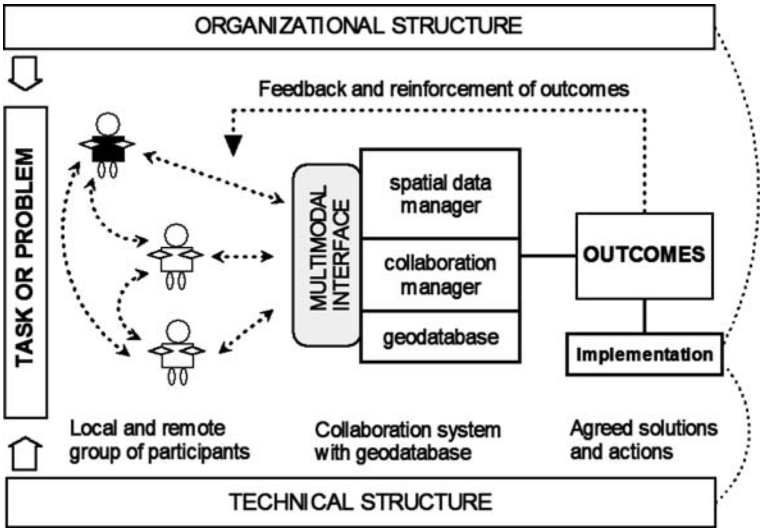
Collaborative geographic information systems; CGIS; Computer supported cooperative work; CSCW; Group spatial decision support systems; GSDSS; PSS; Planning support systems

Definition

Geocollaboration is an emerging area of study examining how spatial information and communication technologies can be designed and adapted to support group interactions that use geographically-referenced data and information [1]. These group interactions normally focus on tasks such as spatial data access and exploration, problem-solving, planning, and decision-making. In a recent classification of knowledge areas within geographic information science, geocollaboration has been interpreted as a specific implementation of group spatial decision support systems (GSDSS) which in turn forms a component of GIS and Society research [2]. In order to support collaborative interactions, group participants need to be able to browse, explore and query spatial data and information. Further, participants must be able to represent knowledge and communicate with each other towards achieving well defined objectives or goals. Hence, key issues of geocollaboration include group support methods such as real time conferencing and sketch mapping, distributed computing using the Web and local area networks, and information communication with maps and scientific visualizations tools [1,3]. Figure 1 shows a general system architecture for geocollaborative interactions consisting of: a group of participants arranged in various configurations of place and time; a computer system for handling geospatial data and group interactions; technical expertise to integrate the system; and organizational expertise to focus the goals of the collaborative process for appropriate implementation.

Historical Background

Geocollaboration as a concept was formally proposed around the year 2000 by researchers from Penn State University (USA) as a focused response to the need for designing and adapting geospatial technologies for supporting group interactions [4]. Geocollaboration represents an important confluence of methods and tools. The need for group based spatial technologies was formally recognized and systematically discussed during the September 2005 Specialist Meeting of the US National Center of Geographic Information Analysis (NCGIA) research initiative on collaborative spatial decision making (CSDM). The CSDM initiative (I-17) investigated the design of interactive group environments for spatial decision making. The research that followed in this area focused on specific issues including structuring the collaborative process [5], embedding spatial data directly into group discussions [6], analyzing map-based group data [7], ensuring



Geocollaboration, Figure 1 General system architecture for geocollaborative interactions

democratic levels of group participation [8], and multiple visual data representation for improved understanding [9]. Efforts on specific aspects of spatial collaboration resulted in many flavors of collaborative system designs such as Spatial Understanding Support Systems, Planning Support Systems, and Collaborative Geographic Information Systems [2]. These systems have their foundations in mainly decision analysis theory, group structuration theory, and geographic information systems and science. Geocollaboration, however, has its foundations in geographic information science, computer supported cooperative work, and distributed computing. With this foundation, geocollaboration seeks to address the impact of technological, social and cognitive factors on group based interactions with geospatial data [1].

Scientific Fundamentals

Figure 1 shows a typical architecture for geocollaborative interactions. Before the system is constructed, well defined tasks or problems to be solved are usually specified by participatory groups and decision makers. A choice amongst the four combinations of same place or time and different place or time designs determines if the collaboration infrastructure will incorporate the Web as a distributed communication medium. Structuring the collaborative interactions follows, and is influenced by the collaborative process focus on either knowledge construction, problem solving, task implementation, data exploration, spatial analysis, decision making, or training. Many structuring approaches are available including shared workspace, argumentation mapping, spatial Delphi, real time conferencing and sketch maps. User interfaces and their designs can vary from a range that includes simple map sketching with digital pens and sophisticated three-dimensional head mounted displays to immersive virtual environments. The capabilities of user interfaces to aid visualization and natural data interactions are constantly being enhanced. These user interfaces connect directly to a distributed or localized computer system that manages the collaborative interactions and geospatial data storage. There are usually three components: a spatial data manager to facilitate data requests and transactions, a collaboration manager to track and monitor the interactions among the group of participants, and a geospatial database to provide spatial data and information as per the request of the spatial data manager component. The collaborating participants eventually generate spatial outputs in the form of maps, scientific visualization products, or geo-referenced data attributes in response to the specified system tasks or problem definition. Iterative analysis of the outputs can generate more robust final outcomes. The geocollaboration system must be embedded in an organizational and technical structure for continued development and support of the local and remote collaborations [6]. Interface design (user-friendly interactions), social dynamics (level of social representation), distributed GIS (equal participation opportunities), and cartography (representation and understanding of reality) principles all have an influence on geocollaboration.

Key Applications

Principles of geocollaboration are applied in many diverse knowledge domains where there is a need to integrate people, technology and data in a spatial context. These applications cover a diverse range from the environmental sciences to engineering, but are more dominant in geography where geospatial data are widely collected and analyzed.

Geography

In geography, geocollaboration principles focus mainly on the people component of the people-technology-data integration. The development of group support methods are of specific interest. Applications for public participation, transportation, and health are outlined.

Public Participation Geocollaboration principles can be applied to designing systems that improve the public's access to information and their contributions to planning and decision making forums. These public interactions usually occur at the same-time/same-place or same-time/different-place [10].

Transportation Transportation planners can use geocollaboration principles to develop systems that reconcile multiple interests and organizational goals. Candidate sites for transportation development can be identified and assessed more effectively for more appropriate spending of public funds [11].

Health In health care and disease management, geocollaboration designs can be used to develop highly interactive systems that allow the public to identify and locate disease incidences thereby allowing more targeted responses from health care professionals.

Engineering

In engineering, geocollaboration principles focus mainly on the technology component of the people-technology-data integration. The development of more user friendly spatial technologies is of particular interest. Applications in disaster management, user interfaces, and distributed databases are outlined.

Disaster Management During times of disaster, the close coordination of resources from multiple organizations is necessary for mitigating and managing the crisis situation. Geocollaboration principles allow for the design and development of systems that can integrate managers, task groups, data resources and collaborative interactions for real-time planning and coordination among teams [12].

User Interfaces Geocollaboration principles are used in the design of natural user interfaces for more embedded interactions and manipulation of geospatial data [13].

Distributed Databases Geocollaboration designs can be used to develop highly accessible and interactive knowledge systems that allow the non-technical individuals to input data into existing spatial databases to improve understanding and awareness of existing conditions [14].

Environmental Sciences

In the environmental sciences, geocollaboration principles focus mainly on the spatial data component of the people-technology-data integration. Developing more accessible spatial databases and providing for greater collaborative data analysis and modeling are of specific interest. Applications in land use, water resources, and forestry are outlined.

Land Use Geocollaboration principles can be used in exploring scenarios and alternative futures for land use planning and analysis [15]. Multiple participants interact with each other and geospatial data towards defining models of reality that best capture common interests, concerns, and goals.

Water Resources Water resource managers can use geocollaboration principles to develop systems that reconcile multiple interests and organizational goals. The influence of technology designs and participation perceptions on the desirable outcomes can be examined experimentally [16].

Forestry The development of participatory model-based forestry tools can benefit from geocollaboration principles for improving model input interfaces and in the visualization of output scenarios.

Future Directions

Any problem or task that requires the integration of people, technology, and geospatial data can benefit from geocollaboration principles in the design of solutions. Applications in national security and human health are new and emerging areas of interest for geocollaboration principles. Collaborative interaction systems with virtual environments can allow military strategists to better simulate command and control operations for training and emergency purposes. Medical personnel can be briefed and deployed in emergencies using distributed collaborative systems that link multiple authorities into a centralized decision-making structure.

On the research frontier, geocollaboration will need to further address key issues including:

- Integrating real-time spatial data from multiple sources and with multiple errors
- Using the Internet as a robust medium to support collaborative communication
- Developing open source toolkits to encourage geocollaboration system designs
- Designing more effective multi-modal interfaces to improve end-user experiences

The application potential and research needs of geocollaboration make it an exciting and rewarding area for study and future enhancements.

Cross References

► Decision-Making Effectiveness with GIS
► Environmental Planning and Simulation Tools
► Multicriteria Decision Making, Spatial
► Web Mapping and Web Cartography

Recommended Reading

1. MacEachren, A.M., Brewer, I.: Developing a conceptual framework for visually-enabled geocollaboration. Int. J. Geogr. Inf. Sci. **18**, 1–34 (2004)
2. Balram, S., Dragicevic, S.: Collaborative Geographic Information Systems: Origins, Boundaries, and Structures. In: Balram, S., Dragicevic, S. (eds.) Collaborative Geographic Information Systems, pp. 1–22. Idea Group Publishers, Hershey, PA (2006)
3. Schafer, W.A., Ganoe, C.H., Xiao, L., Coch, G., Carroll, J.M.: Designing the next generation of distributed, geocollaborative tools. Cartography Geogr. Inf. Sci. **32**, 81–100 (2005)
4. MacEachren, A., Brewer, I., Cai, G., Chen, C.: Visually enabled Geocollaboration to support data exploration and decision-making. In: Proceedings of the 21st International Cartographic Conference, 10–16 August: Durban, South Africa (2003)
5. Jankowski, P., Nyerges, T.: Geographic information systems for group decision making: Towards a participatory geographic information science. Taylor and Francis, New York (2001)
6. Dragicevic, S., Balram, S.: A Web GIS collaborative framework to structure and manage distributed planning processes. J. Geogr. Syst. **6**, 133–153 (2004)
7. Jankowski, P., Andrienko, N., Andrienko, G.: Map-centred exploratory approach to multiple criteria spatial decision making. Int. J. Geogr. Inf. Sci. **15**, 101–127 (2001)
8. Ghose, R.: Use of information technology for community empowerment: Transforming geographic information systems into community information systems. Trans. GIS **5**, 141–163 (2001)
9. Shiffer, M.J.: Multimedia GIS for planning support and public disclosure. Cartography Geogr. Inf. Syst. **25**, 89–94 (1998)
10. Kingston, R., Turton, I., Carver, S., Evans, A.: Web-based public participation geographical information systems: An aid to local environmental decision-making. Comput. Environ. Urban Syst. **24**, 109–125 (2000)
11. Nyerges, T.L., Montejano, R., Oshiro, C., Dadswell, M.: Group-based geographic information systems for transportation improvement site selection. Transp. Res. Part C-Emerging Technol. **5**, 349–369 (1997)
12. Cai, G.R., MacEachren, A.M., Brewer, I., McNeese, M., Sharma, R., Fuhrmann, S.: Map-mediated GeoCollaborative Crisis Management. In: Intelligence and Security Informatics, Proceedings, vol. 3495. pp. 429–435 (2005)
13. MacEachren, A.M., Cai, G., Brewer, I., Chen, J.: Supporting map-based geocollaboration through natural interfaces to large-screen displays. Cartographic Persp. **54**, 16–34 (2006)
14. Mason, B., Dragicevic, S.: Web GIS and knowledge management systems: An integrated design for collaborative community planning. In: Balram, S., Dragicevic, S. (eds.) Collaborative Geographic Information Systems, pp. 263–284. Idea Group Inc. (2006)
15. Klosterman, R.E.: Computer applications in planning. Environ. Plann. B: Plann. Des. **25**, 32–36 (1998)
16. Nyerges, T., Jankowski, P., Tuthill, D., Ramsey, K.: Collaborative water resource decision support: Results of a field experiment. Ann. Association Am. Geogr. **96**, 699–725 (2006)

GeoDa

► Data Analysis, Spatial

Geodemographic Segmentation

Austin Troy
Rubenstein School of Environment and Natural Resources, University of Vermont, Burlington, VT, USA

Synonyms

Neighborhood segmentation; Spatially agglomerative clustering; Geographic market segmentation; Regionalization, spatial ontologies; Cluster analysis; Partitioning; Regression, geographically weighted

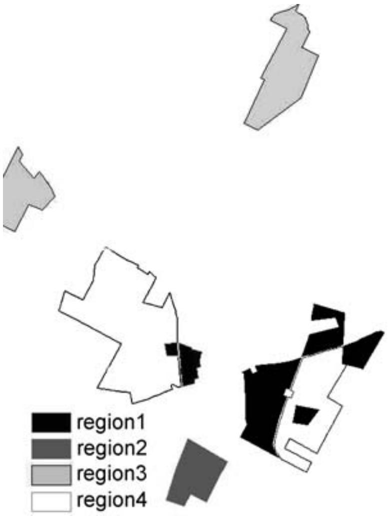
Definition

Geodemographic segmentation refers to a range of methods used for classifying and characterizing neighborhoods or localities based on the principal that residents living near each other are likely to have similar demographic, socio-economic and lifestyle characteristics. It is used for a wide range of purposes including direct marketing, retail location, service area analysis, housing market analysis, and public service targeting.

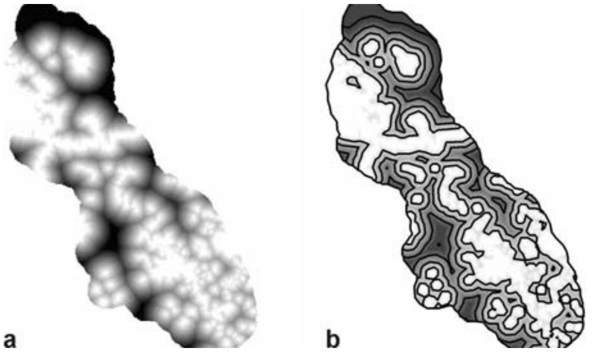
Many methodological approaches exist for geodemographic segmentation. Some form of cluster analysis is generally used to assign entities (e. g. zip codes, property parcels, neighborhoods) to classes which can eventually form larger objects, or segments. Entities are classified so as to minimize heterogeneity within groups relative to between groups. The classification taxonomy can incorporate any number of classes and can also make use of nested or non-nested hierarchies (see ► Hierarchical Spatial Models, this volume). The output is the class membership for each constituent object. While cluster analysis is generally aspatial (although geographic coordinates can be included as additional dimensions), adjacency or connectivity rules can be applied afterwards specifying when entities of the same class can be merged. The geographic nature of segments depends on the entities they are built from. Where segments are built from spatially exhaustive polygons, the

G

Geodemographic Segmentation, Figure 1a and b Aggregation of spatially exhaustive subunit polygons into spatially exhaustive segments



Geodemographic Segmentation, Figure 2 Segmentation of "island polygons"



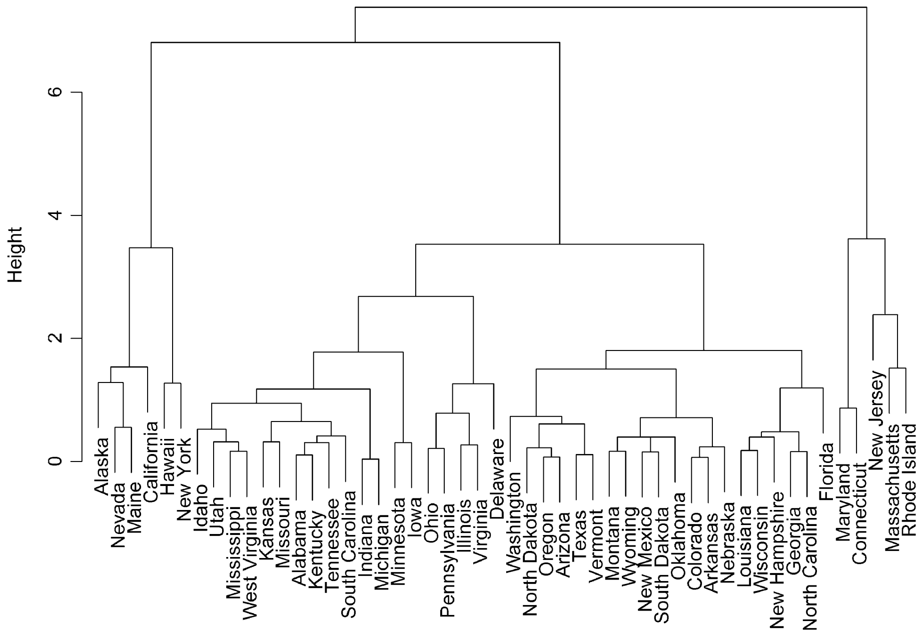
Geodemographic Segmentation, Figure 3 Segementation of continuous fields

output may be a landscape of larger spatially exhaustive polygons (Fig. 1a and b). Where they are built from spatially isolated (i. e. "island") polygons, the output may be new multi-part regions whose constituents do not necessarily touch (Fig. 2). Point objects or raster surfaces (Fig. 3a and b) may also be segmented.

Historical Background

The history of cluster analysis dates back to the 1930s, with early works such as R.C. Tryon's *Cluster Analysis* [1], but the greatest advances in the technique's application and development, and much of the literature, comes from the 1960s and after. One of the early landmark texts on cluster methods was *Principles of Numerical Taxonomy*, written by Sokal and Sneath in 1963 [2]. Neighborhood and geodemographic segmentation, dates back even earlier, to the sociologist Charles Booth who, in the early 20th century, developed a multivariate system of classifying neighborhoods to study patterns of poverty in London. In the 1920s sociologists from the "Chicago School," such as Robert E. Park and Ernest Burgess built on this foundation shortly thereafter with a theory of urban "natural areas," which posited that competition for land in cities among social groups led to its division into geographic units similar to ecological niches [3]. Their application of this theory involved considerable pre-digital overlay analysis of maps, including the *Social Science Research Base Map of Chicago* which combined all available spatially referenced data about the city and was used to detect boundaries of the "zones" that their theory predicted. Geodemographic analysis became increasingly sophisticated with the availability of electronic Census data, the enumeration of smaller Census units, such as block groups, and advancement of cluster analysis.

The commercial application of geodemographics has its roots in 1961, when Jonathan Robbin formed the General Analytics Company, a marketing research firm that began researching geodemographic methods. This firm was pur-



Geodemographic Segmentation, Figure 4 An agglomerative hierarchical cluster tree grouping states based on population density and owner occupancy rates

chased and later re-acquired by Robbin in 1971, at which point it became Claritas Inc., the company that created the PRIZM®, the first major geodemographic segmentation, in 1978. It uses Census and other data to classify US Census Block Groups into hierarchically nested segments. This and later products, such as Mosaic by Experian and ACORN™ (A Classification Of Residential Neighbourhoods) by CACI, are used to help target marketing efforts, site retail locations, and generally understand how purchasing behavior varies by neighborhood. Many of the key concepts of geodemographic segmentation in the post-digital age are summarized by Micheal Weiss in a trilogy of books from 1988, 1994 and 2000 [4,5,6] as well as by Debenham et al. [7].

Scientific Fundamentals

Segmentation is fundamentally a classification exercise. In the simplest case, a typology is developed a priori with set criteria for membership in each class. Spatial entities (e. g. zip code, city, house, etc.) are then classified based on meeting these criteria. More commonly, segmentation uses a posterior approach in which classifications are based on cluster analysis or other multivariate methods. Cluster analysis is a data structuring tool that is generally used as an exploratory rather than confirmatory tool. It organizes data into meaningful taxonomies in which groups are relatively homogeneous with respect to a specified set of attributes. That is, it maximizes the association between objects in the same group while minimizing the association between groups. It does so based on the concepts of dissimilarity or distance in *n* dimensional space, where the axis of each dimension represents some attribute. While the nomenclature of cluster distance sounds spatial, in fact the distances being measured do not represent real space, but rather attribute space.

There are a number of clustering techniques that can be used in geographic segmentation. These can be broadly broken up into partitioning and hierarchical methods. Hierarchical methods work by either dividing (divisive methods) or aggregating (agglomerative methods) groupings of data using a hierarchical structure. Agglomerative clustering starts by treating every data observation as a separate group in itself and then groups those observations into larger groups until there is a single group (Fig. 4). That is, it sequentially lowers the threshold for defining uniqueness, causing progressively fewer observations to be seen as "dissimilar." Dissimilarity or similarity is based on the "distance" between observations, which is represented in the "height" axis in Fig. 4. In multi-dimensional attribute space, distance refers to how dissimilar the attributes of an observation are from another. When attribute variables are in different numeric scales, it is often required to standardize the data so that no one variable is overly weighted. Distance can be measured as Euclidean (straight line) distance (Eq. 1), squared Euclidean distance (Eq. 2), Manhattan (city block) distance (Eq. 3), and Chebychev distance (Eq. 4), among many other approaches.

$$\mathrm{distance}(x, y) = \Sigma_i (x_i - y_i)^2 \tag{1}$$

$$= \{\Sigma_i (x_i - y_i)^2\}^{1/2} \tag{2}$$

$$= \Sigma_i |x_i - y_i| \tag{3}$$

$$= \mathrm{Max}|x_i - y_i| \,. \tag{4}$$

This approach is useful for *a posteriori* data explorations and allows users to interpret how clusters relate to each other based on the patterns in which they branch. Divisive clustering is the opposite of agglomerative clustering; it starts with all data in a single set, which is successively divided into smaller groups until each group is a single observation.

Partitioning methods are often used when the analyst has some a priori notion about how many or what type of clusters to expect. The clusters can then be analyzed for systematic differences in the distributions of the variables. Partitioning iteratively creates clusters by assigning observations to the cluster centroid that is nearest. The most common partitioning method is k means, popularized by Hartigan [8], which works by randomly generating *k* clusters, determining the location of the cluster center, assigning each point to the nearest cluster center, and then iteratively recomputing the new cluster centers until convergence occurs, which is generally signaled by when point-cluster assignments are no longer changing. Other partitioning approaches use different measures of dissimilarity. For instance, Expectation Maximization (EM) Clustering maximizes the probability of the data given the final clusters by analyzing the data's probability distributions to compute probabilities of cluster memberships. This method is frequently used for market and geodemographic segmentation because it can be applied to both continuous and categorical variables, unlike k means. Yet another partitioning method is partitioning around medoids (PAM) [9]. Rather than minimizing distances, as k means does, PAM minimizes dissimilarities, a more robust measure of difference measured in a dissimilarity matrix. The dissimilarity matrix allows PAM to perform clustering with respect to any distance metric and allows for flexible definition of distance. The use of medoids rather than centroids makes PAM less sensitive to outliers. Finally, PAM has the advantage of producing several outputs that help in assessing the strength of classifications. In particular, silhouette plots and scores (which range from -1 to 1) give an idea of how well an observation is grouped. A score of 1 means it is well classified, -1 poorly classified, and zero means it lies between two clusters. A clusterplot provides a further diagnostic, showing how much dimensional overlap or isolation there is between clusters. These outputs allow one to run PAM several different times with different combinations of variables or different numbers of clusters, to assess how the strength of groupings changes.

A new set of partitioning methods was developed in the nineteen nineties, based on Neural Network (NN) and Artificial Intelligence paradigms [10]. NN essentially uses a nonlinear and flexible regression technique which does not require prior assumptions of the distribution of the data to classify data. NN methods have the advantage of evaluating similarities based on a set of multi-dimensional criteria, as opposed to traditional clustering algorithms which generally use a single measure of dissimilarity. However, they suffer from the problem of some non-hierarchical clustering algorithms of requiring a user defined number of clusters and random starting points in *N* dimensional space when seeding the clustering algorithm. Hence, there is no clear superior method. One study found that while NNs did not perform superior to traditional k-means classifiers in geodemographic segmentation, a combination of the two, with NNs generating the input seed for k-means resulted in superior classification [11].

Yet another approach to clustering is Multivariate Divisive Partitioning (MDP). Claritas, Inc. used a form of MDP called Classification and Regression Tree (CART) analysis to more functionally classify their market segments. In this more supervised form of clustering, an analyst chooses a dependent variable or behavior they wish to model and then conducts a stepwise process to determine which variables, and which breaks in the values of those variables, best divides a single segment into two segments with the greatest difference in that behavior. Splitting then continues iteratively until a threshold of similarity in the dependent variable is reached.

Where there are a large number of variables being considered, segmentation often makes use of Principal Components Analysis (PCA), a data reduction tool which condenses a large set of variables into a smaller number of discriminating factors, while eliminating ineffectual noise. These factors are represented using standardized linear combinations of the original variables. Generally, most of the original variation in the variables is explained in the first principal component. Because each component is orthogonal, each subsequent component should be uncorrelated with the previous one, and hence explain less variance. Thus, while the number of principal components is equal to the number of variables, only a few of the principal components need be used because they explain most of the variance. Segmentation uses PCA to condense the vast number of socio-economic and demographic variables that differentiate neighborhoods into a much smaller set of largely uncorrelated factors, or principal components. For example, Applied Geographic Solutions (AGS) uses PCA in building its MOSAIC™ geodemographic segmentation. The user's guide for this product states that PCA is an excellent approach because it makes it easier to identify and isolate "site signatures," or combinations of traits that make neighborhood types unique, since each factor is a condensed and distilled representation of many related variables, and each factor is theoretically independent.

Geodemographic Segmentation, Figure 5 Adjacent polygons with the same cluster value are dissolved into single polygons

Once principal components have been generated, observational units can be classified by cluster analysis of these values or through simple numeric breaks.

The methods described up to this point are aspatial. While the observations being analyzed may represent spatial entities (e. g. zip codes or watersheds), cluster analysis does not explicitly consider spatial relationships unless specified. However, it follows logically that when spatial autocorrelation is present in the variables being analyzed, nearby geographic entities will be more likely to have the same cluster value than distant entities. That is because spatial autocorrelation also describes similarity in attribute values, just as cluster analysis does. Regardless of autocorrelation, however, actual spatial variables, such as latitude and longitude coordinates, can be used as variables in cluster analysis.

Even if no spatial variables are used in cluster analysis, spatial rules can be applied afterwards. For instance, a cluster analysis might give a result where very distant entities have the same cluster designation. In this case the analyst may wish to enforce a rule that spatial entities with the same cluster designation that do not touch or are not within a certain distance have different cluster designations. An analyst may also wish to combine entities with the same cluster designation if they are adjacent or within a certain distance. In the case of polygons this can easily be done by dissolving common boundaries to form a single polygon from multiple polygons with the same value, as shown in Fig. 5, or aggregating non-adjacent polygons into topological regions.

The output of segmentation will by definition depend on the spatial nature of entities or observations. Where spatially exhaustive polygons (e. g. zip codes) are used as the initial observational units, the cluster analysis yields a spatially exhaustive landscape classification. While adjacent polygons of the same class can be merged, the end result gives a class for all locations where land is present. Hence there is no need for any interpolation. In some cases, polygons are not actually exhaustive, but are functionally so, such as in the case of property parcels, which include all land except for public rights of way, which likely are irrelevant to most analyses using segmentation. Points may also be classed into segments although by definition the resulting map would not be geographically exhaustive. If one desires to segment a landscape from point data, though, some form of interpolation is necessary. That is, where no areal boundaries exist, they must somehow be detected. In the simplest case, Thiessen polygons/Voronoi diagrams (see ► Voronoi Diagram, this volume) are created around the points to form an exhaustive landscape, with those polygons inheriting all of the overlaying points' attributes. Polygons are then segmented through cluster analysis. In the continuous surface approach, interpolated surfaces are created for each variable relevant to the segmentation. Cluster analysis is then performed on the multiple values associated with each grid cell to yield a cluster designation. Due to the spatial autocorrelation associated with interpolation methods, nearby cells will be likely to be in the same class and in fairly large contiguous areas of the same class, which can then be converted to polygons to form an exhaustive landscape. As described under "housing market segmentation" below, this problem is very relevant to the study of housing markets, where data often come in household-level point format, but where exhaustive regions are desired as output.

Key Applications

Marketing

The most significant application of geodemographic segmentation is for marketers. Marketing is most effective when targeted at the audience most likely to purchase a product or service. Market segmentation products help companies target where and how they spend their marketing and advertising money by comparing the profile of their customer base with the profile of a particular geographic region. This can also help marketers successfully replicate their efforts and expand their campaigns.

A number of proprietary segmentation systems have been developed for this purpose, including Experian's MOSAIC™, Claritas's PRIZM®, and Consolidated Analysis Centers Incorporated's ACORN™. These traditionally use Census data, as well as other ancillary data, such as point-of-sale receipts data, purchasing patterns, mortgage and financial information, and household surveys. Claritas now uses household-level data, including responses from lifestyle surveys. In the United States, most of these systems use Census block groups, which were designed to approximate "neighborhoods" of about 350 households, as the minimum geographic unit of analysis, although the recently released PRIZM® NE used an algorithm to attempt to interpolate PRIZM® segments to individual households. In the United Kingdom, post codes are often used as the minimum mapping unit. Each segment is expected to be relatively homogeneous in its purchasing habits. Segment designations are generally updated periodically to reflect changes in the neighborhood.

Many segmentation systems are hierarchical as well. For instance, PRIZM® NE has three hierarchically nested levels of segment classes. All observational units are first classed into one of 5 "urbanicity" classes (PRIZM® 5), representing degree of urbanization. Those 5 classes are further broken down by socio-economic status to yield 14 "social groups" (PRIZM® 14). Those are then further broken down by lifestyle factors to yield 66 final segments (PRIZM® 66). Each segment and segment aggregation is given a unique name and description. So, for instance, the "Urban" PRIZM® 5 class contains a number of PRIZM® 14 sub-classes, such as "Urban Midtown." Within that are three PRIZM® 66 sub-sub-classes, including "Urban Achievers," "Close in Couples," and "Multi-Culti Mosaic." Detailed socio-economic and demographic generalities are given about each segment, as well as information about their consumption preferences. For instance, "Urban Achievers" tend to have no children in the household, be younger, and pursue active social lives, including going to clubs, bars, and concerts. More specifically, they are more likely to eat pita bread, shop at Nordstroms and Banana Republic, watch tennis, listen to jazz, and go downhill skiing.

Retail Site and Performance Modeling

When retail facilities choose a location, they consider the size and characteristics of the consumer population within a service or trade area. Geodemographic segmentation systems are commonly used for this purpose, especially by companies with large numbers of outlets. Specifically, segmentation systems can be used to assess the penetration rate of different market segments targeted by the retailer. This approach is often combined with service area analysis, in which a network algorithm is used to delineate the geographic region that is within a certain drive time of the facility (the maximum drive time for a facility will depend on the nature of the facility, e. g. home improvement store, versus convenience store). Once this region is established, the number of households belonging to each segment type can be estimated. Sites can be found that maximize the number of households belonging to the segments that are likely to demand the particular good or service. Gravity models can be used to account for competing facilities in the service area. Further, this approach can be used to help stores fine-tune what product lines they should feature based on the characteristics of the surrounding neighborhoods. For instance, a grocery chain may stock a certain store with more organic produce and high-end gourmet lines based on surrounding market segment. This approach can has been used for large facilities like big box store, smaller facilities like coffee shops, and even automated teller machines (ATMs).

Retail stores have also been segmented themselves based on their characteristics and the characteristics of their trade area. Kumar and Karand [12] created such a segmentation for grocery stores, intended to account for the interaction between internal store attributes, such as "scrambling of service," and trade area characteristics. They found that such a segmentation did predict differences in retail performance.

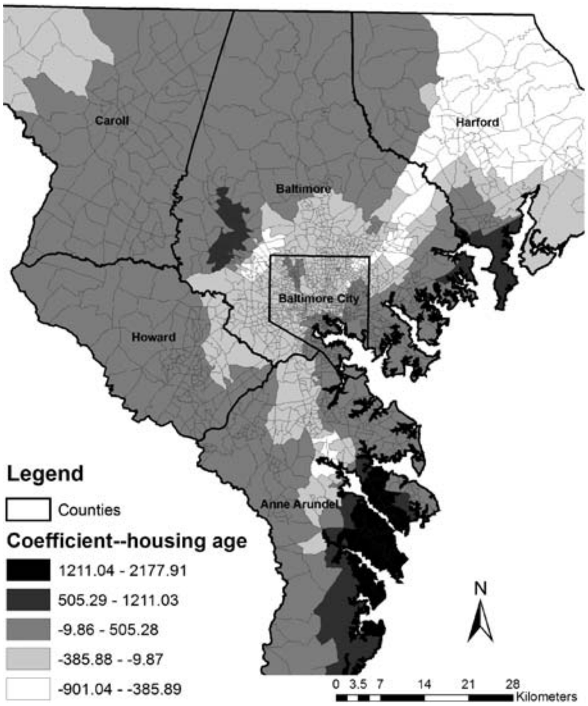
Public Service Planning

Public agencies can use geodemographic approaches to help target the location and scale of public facilities, based on the same principals of retail site modeling, but substituting public need for profit maximization. In some cases this may involve searching out demographic segments that are not well served in the private sector. In others it may involve matching facilities to areas with a certain profile expected to use those facilities. The potential applications range from siting of public transportation facilities, to libraries, to youth centers.

Housing Market Analysis

Economists contend that there are distinct housing "submarkets" within each urban area. These represent areas with relative similarity in housing characteristics, like price, size, type, ownership status, and quality of housing, among other variables. Some also contend that submarkets are defined by the relationship between these variables and price [13]. Accounting for housing submarkets through segmentation can result in great improvements for several applications, including derivation of price indices, automated valuation models (AVM's) or mass appraisal methods for mortgage lenders, and studies of non-market goods using hedonic analysis (in which property price is regressed against explanatory variables in order to disaggregate the contributors to housing price and place a value on characteristics that are not directly priced in the market). That is because these methods all rely on the assumption of a homogeneous housing market. When housing submarkets exist and are accounted for separate models can then be estimated for each market segment, resulting in increased precision. Segmentation has also been found to be necessary for generating welfare measures of benefits from hedonic indices [14,15].

A variety of methods have been proposed for segmenting housing markets, including both geographic and non-geographic methods. There has been a wide range of opinions as to whether housing markets are defined by spatial criteria or not. In some studies, individual non-contiguous properties within the same neighborhood have been classed into different segments based only on the internal characteristics of those properties and disregarding location [16], while in many others geographic partitions have been used. One recent study Bourassa et al. compared submarkets defined by small spatially contiguous geographical units (based on real estate appraisers' data) versus submarkets defined based on the properties' internal characteristics only, disregarding spatial contiguity, and found that stratifying data by the former resulted in more accurate hedonic housing price equations [17]. As the authors point out, housing submarkets matter and it is geography that makes them matter. For those using geographic units to segment housing markets, some use exogenously derived geographies, like Metropolitan Statistical Areas [14] or real estate board jurisdictions [13], while others derive segment boundaries statistically.



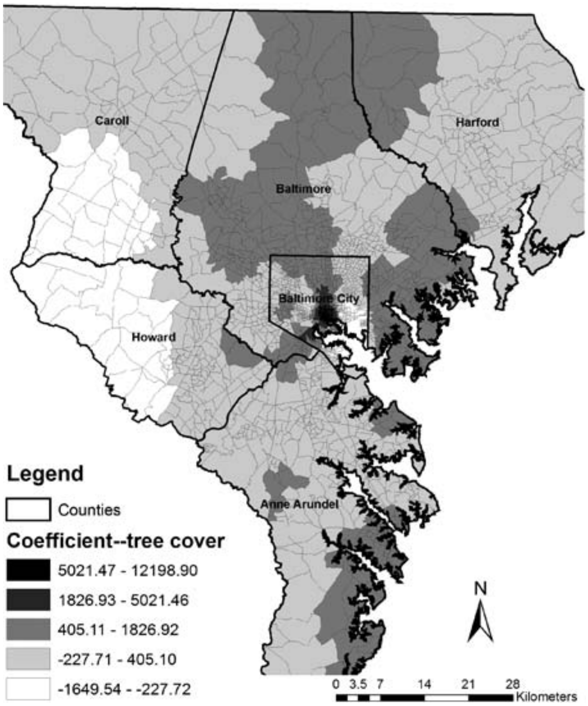
Geodemographic Segmentation, Figure 6 Block groups with grayscale values graduated by GWR coefficients for median house age as a predictor of housing price

Urban and Environmental Planning

Just as geodemographic segmentations help marketers target their clientele, they can also help planners target investments or policies. A recent study found that PRIZM® lifestyle clusters could actually be used to predict differences in urban public right of way vegetation, suggesting that PRIZM® class could be used to target future street tree investments. Other potential applications include targeting outreach efforts towards residential areas that are expected to produce higher levels of non-point source pollution, targeting pedestrian improvements to districts where residents are predicted to commute by foot or bike, helping decide the type of recreational facilities to invest in for neighborhoods based on expected preferences, and locally adjusting parking requirements based on expected car usage by neighborhood.

Future Directions

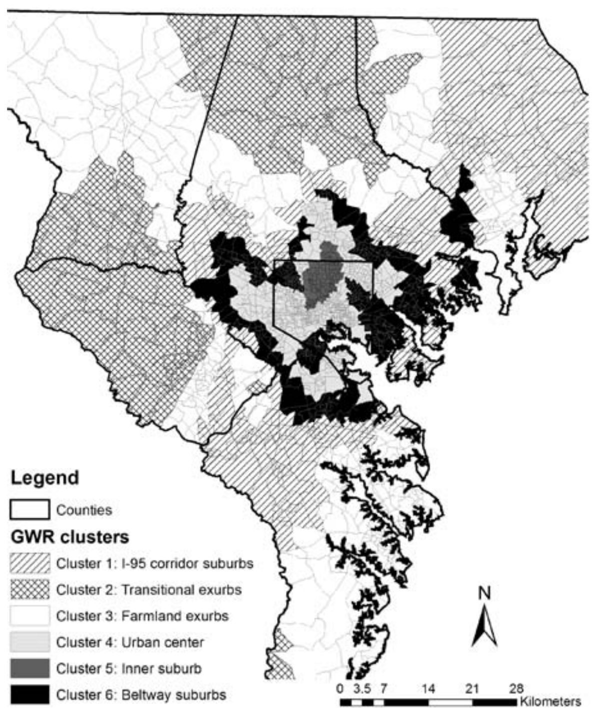
A potentially more sophisticated but rarely used approach towards segmentation is based on the use of Geographically Weighted Regression (GWR; see ► Spatial and Geo-



Geodemographic Segmentation, Figure 7 Block groups with grayscale value graduated by GWR coefficients for mean tree cover as a predictor of housing price

graphically Weighted Regression, this volume) [18]. This approach could be useful in marketing when there is some factor that cannot be easily quantified (e. g. local taste, values, customs) but which mediates the relationship between geodemographic variables (e. g. income, education) and sales. It may also be helpful in defining spatial housing market segments by showing where the changes occur in the relationship between price and attributes.

GWR is a spatial regression method which tests whether model coefficients are non-constant over space, unlike traditional regression which assumes spatial stationarity (that is, the relationship between dependent and independent variables is constant in space and time). In this method, separate regressions are run centered on each observation, with a spatial kernel determining which observations are included in the population of each localized regression and how they are weighted, based on a distance decay function. An adaptive or fixed size kernel can be used to determine the number of local points that will be included, depending on the spacing of the data. The GWR model is tested against a stationary regression model to see if its increased complexity is warranted relative to the increased predictive power. Individual parameters can then be tested for nonstationarity. Those that are can be analyzed by looking at spatial patterns in the distribution of coefficient or test statistic values, which vary continuously across the landscape. In other words, the visual pattern shows how the relationship between the independent and dependent variables is spatially variable. For instance, in certain parts of a city, the presence of parks may be positively associated with property value (where parks are well-maintained) and in other areas it may be negatively associated (where parks are characterized by crime). In other words, the spatial variation in the coefficients may signal important omitted variables. Figures 6 and 7 illustrate this approach for a GWR predicting mean housing price by Census block group; the color is graduated by the GWR coefficient for median house age in Fig. 6 and tree canopy cover in Fig. 7, holding all else equal. As these figures show, the relationships between housing age and price and tree canopy and price are in some places negative and some places positive. Cluster analysis can then be performed on the coefficient values to classify each observation. Because nearby observations will tend to have highly correlated coefficient values (since they are likely to have be part of the same moving-window estimation regression subset), this means also that nearby observations are likely to be classified similarly.



Geodemographic Segmentation, Figure 8 Block groups segmented into clusters representing housing market type using the GWR method

This approach can also be used on any geographical unit. Figure 8 shows the results of an analysis in which median housing value at the block group level was regressed against a number of socio-economic and spatial predictor variables using GWR for the Baltimore, MD metropolitan region. The resulting coefficients were analyzed with cluster analysis and, based on this, each block group was classified into one of six segments. As this map shows, individual members of each segment are generally spatially contiguous because of the spatial autocorrelation of GWR coefficients.

Cross References

► Hierarchical Spatial Models
► Spatial and Geographically Weighted Regression
► Voronoi Diagram

Recommended Reading

1. Tryon, R.C.: Cluster analysis; correlation profile and orthometric (factor) analysis for the isolation of unities in mind and personality., Ann Arbor, Mich.: Edwards brother, inc., lithoprinters and publishers. viii, 122 p (1939)
2. Sokal, R.R., Sneath, P.H.A.: Principles of numerical taxonomy. Freeman W.H., San Francisco, xvi, 359 p (1963)
3. Park, R.E., et al.: The City. The University of Chicago Press, Chicago, Ill 239 p (1925)
4. Weiss, M.J.: The clustering of America, 1st edn. Harper & Row, New York, xvi, 416 p (1988)
5. Weiss, M.J.: Latitudes & attitudes: an atlas of American tastes, trends, politics, and passions: from Abilene, Texas to Zanesville, Ohio. 1st edn. Little, Brown, Boston, 224 p (1994)
6. Weiss, M.J.: The clustered world: how we live, what we buy, and what it all means about who we are. 1st edn. Little, Brown, Boston, viii, 323 p (2000)
7. Debenham, J., Clarke, G., Stillwell, J.: Extending geodemographic classification: a new regional prototype. Environ. Plann. A **35**(6), 1025–1050 (2003)
8. Hartigan, J.A.: Clustering algorithms, Wiley, New York, xiii, 351 p (1975)
9. Kaufman, L., Rousseeuw, P.J.: Finding groups in data: an introduction to cluster analysis. Wiley series in probability and mathematical statistics. Applied probability and statistics.: Wiley, New York, xiv, 342 p (1990)
10. Schèurmann, J.: Pattern classification: a unified view of statistical and neural approaches. Wiley, New York, xvii, 373 p (1996)
11. Balakrishnan, P., et al.: Comparative performance of the FSCL neural net and K-means algorithm for market segmentation. Eur. J. Oper. Res. **93**(2), 346–357 (1996)
12. Kumar, V., Karande, K.: The effect of retail store environment on retailer performance. J. Bus. Res. **49**(2), 167–181 (2000)
13. Palm, R.: Spatial Segmentation of the Urban Housing Market. Econom. Geogr. **54**(3), 210–21 (1978)
14. Brasington, D., Hite, D.: Demand for environmental quality: a spatial hedonic analysis. Reg. Sci. Urban Econ. **35**(1), 57–82 (2005)
15. Zabel, J.E.a.K.A.K.: Demand for air quality. Sage Urban Stud. Abstr. **29**(1), 3–135 (2001)
16. Lipscomb, C., Farmer, M.: Household diversity and market segmentation within a single neighborhood. Ann. Reg. Sci. **39**(4), 791–810 (2005)
17. Bourassa, S., Hoesli, M., Peng, V.: Do housing submarkets really matter? J. Housing Econ. **12**(1), 12–28 (2003)
18. Fotheringham, A.S., Brunsdon, C., Charlton, M.: Geographically weighted regression: the analysis of spatially varying relationships. Wiley, Chichester, England; Hoboken, NJ, USA xii, 269 p (2002)

Geographic Coverage Standards and Services

WENLI YANG, LIPING DI
Center for Spatial Information Science and Systems, College of Science, George Mason University, Fairfax, VA, USA

Synonyms

Geographic Phenomena

Definition

A geographic coverage is a representation of a phenomenon or phenomena within a bounded spatiotemporal region by assigning a value or a set of values to each position within the spatiotemporal domain. This definition of geographic coverage implies that there is usually more than one value for the each phenomenon type within its spatiotemporal domain. This is in contrast to a discrete geographic feature to which a single value is assigned for each feature type. For example, a coverage representing the flow rate of a river has different flow rate values at different positions of the river, while a discrete feature representing river type has only one type value for the same river. Geographic coverage standards specify schema and frameworks for geographic coverage or coverage components. They provide common languages in describing, publishing, accessing, and processing geographic coverage data. A service is a distinct part of the functionality that is provided by an entity through interfaces [1]. Geographic coverage services are those parts of functionality that are used to access and process coverage data. While in its generic definition a service can be provided through any interface, geographic coverage services are commonly referred to as those that have standard interfaces defined by widely recognized standardization bodies.

Historical Background

The term coverage has been used in Geographic Information System (GIS) software since the early 1980s to refer to a set of data representing the geographic phenomena in certain spatiotemporal regions. Geographic coverage can be conceptually modeled in many different ways. Thus, there are many technical languages that can be used to describe the complex coverage information and the operations that can be performed on a coverage. The lack of common, consensus based modeling on geographic information results in difficulties in exchanging geographic data among different GISs and in collaborating among different GIS software components. In 1994, the OpenGIS project presented a vision of making different GIS processing software systems able to communicate with each other over the Internet by using a set of open, standard compliant interface protocols. An attempt of formally defining geographic coverage was made in one of its draft Open Geodata Interoperability Specifications. In the same year, the Open GIS Consortium Inc. (OGC), which was renamed to the Open Geospatial Consortium in 2004, was launched. Through multiple revisions, a more complete concept of coverage was drawn in 1999 in the OGC's Abstract Specification Topic 6 [2]. Since the late 1990's, the Technical Committee 211 (TC211) of the International Organization for Standardization (ISO) has been working on a number of standards and technical specifications on geographic coverage and coverage components [3], most notably, the ISO19123 standard, "Schema for coverage geometry and functions" [4]. This standard defines a conceptual schema for the spatial characteristics of coverages, including the concept of geographic coverage, its spatial domain characteristic, and the relationship between the spatiotemporal domain of a coverage and its attribute range. In the meantime, service standards either for or related to coverage or coverage components have been developed, such as the OGC's "Web Coverage Service" [5], and ISO/TC211's "Geographic Information-Service" [6].

Scientific Fundamentals

Geographic phenomena can be observed in two forms, one is discrete and the other is continuous. Discrete phenomena are usually objects that can be directly recognized due to the existence of their geometrical boundaries with other objects, such as a road, a lake, and a building. Continuous phenomena usually do not have observable boundaries and vary continuously over space, such as temperature, air quality, and reflected solar radiation from land surfaces. Due to such differences between discrete and continuous phenomena, the methods of describing these two categories of phenomena are also different. A discrete phenomenon can be described by a curve representing its boundary and some values representing its attributes. The same attribute has the same value for the within the boundary. For example, a field of crop can be described by its boundary and attributes such as its area, estimated yield, name of planted crop, and owner of the field. An attribute value, e. g., estimated yield, is the same at any position in the field because the field is described as one single discrete object. The attribute value for a continuous phenomenon, on the other hand, varies from one location to another and, thus, must be recorded at each direct position. For example, the reflected solar energy from a land surface as shown from a satellite image changes continuously across the land surface and, therefore, each pixel value of the satellite image represents the amount of the reflected solar energy from the location where the pixel locates. When information about a geographic phenomenon is stored in a computer system, different data structures are usually used for discrete and continuous phenomena. Information of a discrete phenomenon is usually stored by recording the attribute values for the phenomenon and a series of spatial coordinate values depicting the spatial geometry of the phenomenon. Information of a continuous phenomenon is usually stored by recording a series of attribute values and a scheme describing how the locations of each attribute value can be determined. The primary difference between the two is that the former stores a usually limited number of values for attributes together with typically much more spatial coordinate values for its geometry, while the latter usually stores a large number of attribute values together with usually relatively small numbers of values describing the scheme of locating the often spatially regularly distributed positions. The information on discrete phenomena and continuous phenomenon are often used differently. The operations performed on the two are also typically different. For example, geometrical overlaying and buffering analyses are frequently performed on the former while numerical computing is often conducted for the latter. In many cases, continuous data are used to derive discrete objects such as classifying a satellite image into land cover types. Thus, there are often differences in data structure designs, data encoding approaches, data accessing and processing methods for these two types of geographic phenomena. Traditionally, data structure for discrete phenomena is referred to as a vector data type, while that for continuous phenomena is referred to as a raster data type. However, the raster data structure is not sufficient to characterize the spatiotemporal domain of continuous geographic phenomena, classify different subtypes of the spatiotemporal domain, depict the relationship between the spatiotemporal domain and the attribute value range, and define appropriate operations that can be performed on coverage data. The term coverage, as herein defined, is intended to provide a concept that can comprehensively describe the continuous geographic phenomena and facilitate the design of standard interfaces for services on data recording of this kind of phenomena.

The standards on geographic coverage define conceptual schema for coverage and analyze coverage types and components. For example, the ISO19123 standard defines fundamental characteristics of coverage, including spatiotemporal domain, attribute range, major coverage types, and operations on coverages. These standards provide a common technology language and guide the development of interoperable services on coverage data. A number of services have been available involving geographic coverage standards, most notably the OGC Web Coverage Service (WCS). The WCS is designed to provide users of geospatial data an interoperable way to obtain coverage data across the Internet. Users can utilize any interoperable client to request from WCS servers of coverage data in interested spatial and temporal extent, with user-specified spatial coordinate reference system and spatial and temporal resolutions, and in a user-defined data encoding format. In contrast to the WCS' ability of providing coverage data, the OGC Web Map Service (WMS) allows clients to obtain coverage data as a map to be viewed or overlaid with other geospatial maps, including both coverage and feature maps. Special purpose service standards and services such as image classification service, image exploitation services, and coverage processing services have either been developed or are currently under development. These can provide more specialized coverage data processing capabilities for different scientific and application communities. Users of such services frequently need more specialized knowledge in their respective areas. For example, the OGC Web Image Classification (WICS) provides interoperable ways for users to perform classification of images, but it requires that a user understand which classifier and/or classification scheme is more suitable for his or her specific purpose.

Key Applications

The concept of geographic coverage and the coverage standards and services are mainly used in geospatial information technology, especially in observing, collecting, processing, archiving, cataloging, and publishing of geographical data for providers, and in searching, discovering, accessing, and acquiring of geospatial data and information for consumers. Primarily because most data and information available are either directly or indirectly related with spatiotemporal dimensions, geographic coverage, coverage standards and services find their applications in many science and application domains, such as atmospheric sciences, environmental sciences, forestry, geography, oceanography, decision making, emergency response, and regional panning. Examples of some specific applications are listed bellow.

Software Industry

With the internet has become an indispensable element of scientific research, in applications as well as in daily life, people increasingly demand that more and more information, including geographic coverage information, be freely exchanged in the Web. Software interoperability is the key for such information exchange. Coverage standards and services play important roles in the development of interoperable software associated with the use or processing of geographic information, such as GISs. Vendors of such software can benefit from following well recognized concepts and standard interface protocols in their designs.

Knowledge Discovery and Data Mining

Knowledge discovery and data mining (KDD) involves identifying novel patterns or models in often a large amount of data. KDD is usually conducted in individual data holdings such as a principal investigator's database. Web technology makes it possible to perform KDD across the entire web as long as data and service providers share common interfaces for accessing the data and services. Coverage standards and services provide essential enabling foundations for such perspectives.

Regional Management

Many geospatial data involved in regional management are historically viewed as relatively static, i. e., not varying constantly with time, and are described by discrete features, for example, transportation routes used in traffic management and stream network lines used in public safety management. Many attributes associated to such seemingly static phenomena are actually quite dynamic if examined from an individual position within the phenomena, such as the traffic flow along a transportation route and the water depth along a river. With the advances of observing capabilities such as real-time imaging, high resolution data at individual positions can be collected and processed. Coverage standards and interoperable coverage services can promote the share and use of data with more dynamic attributes and help regional management.

Government Policy

Geographic coverage standards and services assist in the process of making government policies regarding intra-agency collaboration and information distribution and sharing, for example, assessing the benefits of restructuring a local information system into conformance with a federal information system, and evaluating privacy protection for citizens in designing geospatial databases.

Citizen's Everyday Life

Citizens can benefit from geographic coverage standards and services in their daily life, such as virtual reality landscapes from earth observing data, digital museums and galleries, and education and training. In the aforementioned traffic line versus traffic flow example, a citizen is able to find the shortest route from one location to another when the road is viewed as a discrete feature. She is also able to find the shortest time needed to drive between the two locations when traffic flow rate is represented as a coverage having data at each and every point along the road and when the coverage is constantly accessible to her through standard compliant services.

Future Directions

Standardization of technology language and service interface protocols related to geographic coverage is still in its early stage and the effort is an ongoing one. Most current coverage services are not in operational use. Many factors are related to the degree of maturity of standardization and interoperable services, such as status of adopting and implementing specifications, server and client development speed, cross-community communication, easiness in adapting to existing service interfaces of data distribution systems, and government policies. Future directions shall include: stabilization of current implementation specifications, additional coverage processing specifications, promotion of coverage standards and interoperable services among major geospatial data providers, communications to more science and application communities who are employing the use of geographic data, and government enforcing policies.

Cross References

► OGC's Open Standards for Geospatial Interoperability

Recommended Reading

1. ISO/IEC: ISO/IEC, ISO/IEC TR 14252 Information technology – Guide to the POSIX Open System Environment (1996)
2. OGC: OGC Abstract Specification, Topic 6 – The coverage type and its subtypes, Version 4. Wayland, Massachusetts (1999)
3. Di, L.: The development of remote sensing related standards at FGDC, OGC, and ISO TC211. In: Proceedings of IEEE International Geoscience and Remote Sensing Symposium, Toulouse, France, 21–25 July 2003
4. ISO: ISO/TC211, ISO 19123 Geographic information – Schema for coverage geometry and functions (2005)
5. Evans, J. (ed.): OGC, Web coverage service (WCS), Version 1.0.0. Wayland, Massachusetts (2005)
6. ISO: ISO/TC211, ISO 19119 Geographic information-service (2005)

Geographic Data Management

► Privacy and Security Challenges in GIS

Geographic Data Mining

► Geographic Knowledge Discovery

Geographic Data Reduction

► Abstraction of GeoDatabases

Geographic Database Conceptual Modeling

► Modeling with a UML Profile

Geographic Databases

► Spatio-temporal Database Modeling with an Extended Entity-Relationship Model

Geographic Dynamics, Visualization and Modeling

May Yuan
Department of Geography and Center for Spatial Analysis, University of Oklahoma, Norman, OK, USA

Synonyms

Dynamics; Process; Evolution, landscape; Maps, animated; Map, centrographic timeseries; Map, bi-plot; Trend-surface analysis; Time-series correlation graph; Geovista; Agent-based models; Evolutionary algorithms; Mutation

Definition

Generally speaking, *dynamics* is the work of forces (or energy) that produces movement and change in a system and manifests interrelationships among the elements of space, time, and forces. Geographic dynamics centers upon such interrelationships near the surface of the Earth and often across physical, biological, and human domains. While the term, *geographic dynamics* is less popular than terms like *atmospheric dynamics* and *geological dynamics*, the concept of geographic dynamics is deeply rooted in a broad range of geographic studies. Time, like space, is fundamental to time geography, historical geography, cultural geography, environmental geography, biogeography, geomorphology, climatology, hydrology, and many other fields of geography. Spatiotemporal interrelationships among forms, patterns, and processes are central to geographic understanding of physical, biological, and human environments.

Geographic dynamics is multiscalar, multidimensional, and complex. From local and regional to global scales, geographic dynamics is displayed in human activities, urban sprawl, land use and land cover change, transformation of cultural, biological, and physical landscapes, and environmental changes in human dimensions. Geographic processes responsible for these changes are afforded by physical forcing, ecological forcing, anthropogenic forcing, or a combination thereof. Complexity of geographic dynamics emerges when geographic processes interact and drive composite changes that must be explained by all involved processes as a whole. These geographic processes may operate at multiple spatiotemporal scales and across multiple thematic domains. Visualization techniques are critical to inspect the multiscalar, multidimensional, and complex nature of geographic dynamics and in seeking forms and patterns embedded in observations and measurements. The forms and patterns serve as the basis on which to build hypotheses and computational models of geographic dynamics.

Historical Background

Field observations and mapping have been the primary means of studying geographic dynamics, and advances in computing technologies have greatly enhanced visualization and modeling capabilities in the study of geographic dynamics.

As early as 1884, geomorphologist, William Morris Davis developed the Davisian model of fluvial landscape evolution with concepts of dynamic systems used to posit the cyclical nature of erosion in landform development. Visualization was limited to sketches, diagraphs, and graphics to illustrate concepts and stages of landform development. Research interest in the temporal dimension was later incorporated into human and cultural geography. In the early 1950s, cultural geographer, Carl O. Sauer studied agricultural origins and dispersals with emphases on the interactions between physical and cultural environments and historical discourses of the interactions. Sauer's school of cultural–historical geography stressed the importance of maps to record, illustrate, and synthesize geographic patterns and explore geographic processes. Analysis of geographic dynamics in these early geographic studies is mostly qualitative and descriptive.

Besides geography, many Earth science disciplines also embrace the concepts of dynamics in their studies. Geologists, meteorologists, hydrologists, and biologists, for example, all have long histories of investigating how forces interact and result in motion and change in their respective domains of interest. *Earth System Science*, particularly, emphasizes dynamics across scales and subsystems of the Earth (e. g., the geographic system). The past and future possible behavior of the Earth system depends upon positive and negative feedbacks among these processes and human activities. Computer models have been widely developed to address the dynamics in and across the Earth's subsystem and in the Earth system as a whole. For example, there are atmospheric general climate models and ocean general climate models to address climate behavior with emphases on the effects of the atmospheric and oceanic dynamics, respectively. As early as the 1960s, attempts began to couple atmosphere–ocean general climate models to improve predictions of climate change.

In the late 1980s, global impacts of human activities were confirmed through the study of ozone depletion and chlorofluorocarbon (CFC) compounds used in industrial and commercial applications. Increased ultraviolet (UV) exposure results in increases in skin cancer and cataracts in human populations. Research into the interrelationships of human–environmental interactions relies heavily upon visualization and modeling. NASA satellite data from the Total Ozone Mapping Spectrometer (TOMS) show temporal sequences of the ozone hole as a dynamic regional phenomenon in the Antarctic, yet the regional phenomenon has shown long-term global impacts on ecosystems, such as a reduction in yield and an alteration in species competition. Empirical and statistical models have been developed to address various ozone impacts.

Visualization and modeling of geographic dynamics is further leveraged by significant advances in remote sensing and survey technologies along with the astonishing growth of internet and computing technologies in the mid 1990s. The study of geographic dynamics has been transformed from a data-poor to a data-rich environment. New visualization methods are being developed to filter massive amounts of data for pattern detection, and new computational models are being proposed to simulate and predict geographic dynamics. The synergy built upon visualization and computation expands into new modes of scientific practices centering upon data mining (DM) and knowledge discovery in databases (KDD).

Scientific Fundamentals

Visualization and modeling of geographic dynamics is a powerful combination for acheiving insights into geographic understanding. By generating an animated sequence of maps of urban growth in the Detroit region, for example, Tobler [1] invoked "the first law of geography: everything is related to everything else, but near things are more related than distant things."

Animated cartography has been the dominant technique for visualizing geographic dynamics since late 1950s. By animating maps (or images) in a time sequence, how a phenomenon changes over space and time becomes visually evident. There is no shortage of eye-catching animated maps for visualizing climate change, hurricane tracking, population dispersion, land use and land cover change, and many other geographic dynamics. Over the last 30 years, cartographers have been investigating methodological and cognitive issues to promote effective graphical communications about change and facilitate an understanding of processes. [2] summarizes six major research areas in animated mapping: (1) methods for animating time-series data, (2) methods for animating across attributes of data, (3) methods for representing uncertainty and data quality in animated maps, (4) designing effective temporal legends and controls, (5) identifying the visual variables of animation, and (6) methods for temporal interpolation and smoothing of sparse datasets.

Compared to static maps, animated maps provide additional opportunities for and challenges to communicating multidimensional spatiotemporal information. Besides the eight conventional visual variables in static mapping, three dynamic variables (*scene duration*, *rate of change between scenes*, and *scene order*) can be used in the design of map animation to emphasize location, change, and attributes of the geographic phenomenon of interest [3]. The effectiveness of methods and designs for map animation is evaluated by cognitive studies through experiments, interviews, and surveys. Slocum [4] argues strongly that cartographic methods must be iteratively tested within a theoretical cognitive framework and engineering principles.

The design of animated maps should include particular attention to the notion of cognitive friendliness because information load and complexity is much higher in animated maps than in static maps. Ignorance of the effects of the added temporal dimension can induce biases in communication. Based on the theory of human information processing, the human eye–brain system does not instantaneously process patterns from short-term to long-term memories. Therefore, information perceived later in a cartographic sequence in our short-term memory may mask previously viewed patterns that have already entered our long-term memory. Monmonier [5] suggested four animation supplements to bridge the perceptual biases in animated cartography: (1) the centrographic time-series map; (2) the biplot, a joint, two-dimensional representation of time units and places based upon two principal components; (3) canonical trend–surface analysis; and (4) the time-series correlation graph.

Beyond animated maps, geovisualization has emerged as a discipline that connects cartographic principles, multivariate statistics, database systems, data mining, and knowledge discovery. Immersive, interactive, and multimodal software packages have been developed to explore and present dynamic spatiotemporal data. GeoVista is one successful example [6] that provides a suite of toolboxes for spatial data visualization and analysis. The study of geographic dynamics is thriving with the recent surge of integrated visualization–computation approaches. Tobler's urban growth simulation research demonstrated the power of combining visualization and modeling techniques. Hägerstrand's *Time Geography* in the 1970s has been rejuvenated with a growing interest in visualization and geocomputation of human activities [7]. In addition to statistical and mathematical models that describe the behavior or transitions of a system in arithmetic, differential, logistic, or other mathematical terms (such as stochastic models, Markov chains, and Monte Carlo simulation), new approaches to spatiotemporal modeling address a wide range of geographic dynamics that account for local effects, interactions, and complexity in space and time. Examples of popular computational models include distributed modeling, *cellular automata*(CA), *agent-based modeling* (ABM), and evolutionary (or geneitc) algorithms.

These geocomputational models present a new school of empirical modeling in geography. Geocomputational models provide opportunities for thought experiments to shape ideas and build hypotheses. Distinguished from statistical or mathematical models, geocomputational models consider behavior at the individual level and examine emergent properties at aggregation. Common to all geocomputational models is spatial discretization. *Distributed hydrological modeling*, for example, starts with the determination of water flow properties and surface conditions (e. g., amount, flow direction, infiltration, roughness, etc.) at discrete locations. Hydrological patterns (e. g., stream networks) and dynamics (e. g., stream discharge) are aggregated properties of hydrology at these discrete locations. Nevertheless, spatial discretization can go beyond the cell-based approach to allow flexibility in spatial positioning and movement [8].

Geocomputational approaches handle individuality from different perspectives. In addition to distributed modeling, CA and spatial ABM have been broadly applied to urban, ecological, and epidemiological applications. CA emphasize transitional rules for cells to change from one state (e. g., rural area) to another (e. g., urban area) to drive changes in spatial patterns over time (e. g., urban growth). Neighbor effects are often incorporated into the development of transitional rules, such that a cell is likely to become an urban cell when all its neighboring cells have changed to urban cells, and collectively the transition of cells from rural to urban states simulates urban growth. *Spatial* ABM, in contrast, is built upon cognitive mobile decision-making agents that interact with each other and the environment. The collective behavior of these individual agents results in large-scale patterns in space and time [9].

Nevertheless, CA and spatial ABM are complementary bottom-up approaches to revealing geographic dynamics at multiple scales. CA simulate environmental conditions and how the conditions may change over space and time. Spatial ABM mimics the spatial behavior of individuals. CA and spatial ABM together can express geographic dynamics as an interactive system of agents and environments and allows for spatial exploration of dynamics in the context of human–environment interactions [10].

Evolutionary (or genetic) algorithms can add "intelligence" to ABM to allow for adaptability and learning. Spatial evolutionary modeling builds upon Darwinian principles of evolution. Each individual agent possesses a genetic make-up that determines its interactions and responses to the environment and environmental change. In addition, every individual is assigned an objective function and a fitness value as a measurement of the individual's reach to objectives. Evolutionary principles, such as selection, mutation, cross-over, learning, and adaptation are incorporated into the constituents and make-ups of individual agents. The added intelligence enables agents to cope with the environment by changing their behavior as well as to account for multicriteria evaluation of the problems they face. Furthermore, evolutionary algorithms allow for changes in agent composition and population: those who cannot adapt themselves to the environment may decease, while there will be births of new generation of agents from cross-over and mutations. These evolutionary functions bring a higher level of complexity into geographic dynamic modeling by extending dynamics from the individuals in ABM to the communities by considering evolutionary processes. With incorporation of dynamic properties into both agents and the environment, the combination of CA, ABM, and evolutionary algorithms offers an enormous potential to integrate object–field approaches into the investigation of human–environment interactions.

Key Applications

The world is dynamic, and geographic dynamics is central to understanding the biophysical environment, the human environment, and their interactions. No geographic knowledge is complete without a full comprehension of the spatial processes responsible for forms, patterns, relationships and the pathways by which these processes shape the landscape over time. Hence, visualization and modeling of geographic dynamics have a wide range of applications across biophysical and human domains.

G

Weather and Climate

Visualization and modeling of atmospheric conditions over space and time is central to weather forecasting and climate prediction. Unidata (http://www.unidata.ucar.edu) has been developing and maintaining impressive suites of tools for display and analysis of weather and climate data. Space–time algorithms using visualization means have been developed to evaluate the spatiotemporal behavior of mesoscale rainstorms and assessment similarity in rainstorm development in space and time [11].

Land Use and Land Cover Change and Urban Development

CA and ABM techniques have been broadly used to simulate land use and land cover change (LULC) and urban development [12]. Scenarios generated by simulations are instrumental in exploring urban growth and LULC possibilities and drawing insights into potential drivers for planning considerations.

Human Activities, Movement, and Interactions

Underpinned by theoretical frameworks from time geography, modeling human activities, movement, and interaction has been carried out by eliciting and analyzing travel patterns, employment choices, and many other human activities with space and time constraints. Visualization is used to track individual movements through space–time paths and to evaluate the individual's accessibility to locations through space–time prisms [13]. Most existing studies are limited to urban environments. Progress in this research area holds great promises for applications in public safety, emergency preparedness, social program development, and a wider range of urban planning and policy making.

Spread of Species, Wildfire, Diseases, or Pollutants

Visualization and modeling of biological and physical processes expand upon the underlying principles that guide the process development. Different species, for example, have distinctive patterns of movement and migration. Some species are solitary, but others operate in herds. Likewise, wildfires behave differently in grasslands or in forests. The spread of disease and dispersion of pollutants are both constrained by environmental conditions and the nature of the dispersion agents. CA and ABM are common methods to simulate biological or physical spreads with proper rules to capture their distinctive spatial process patterns [14]. Furthermore, modeling of wildfire spread and pollutant dispersion must incorporate the underlying physical mechanisms (e. g., thermal dynamics and atmospheric dynamics) according to scientific understanding.

Future Directions

Geographic dynamics is ubiquitous and essential to advance our understanding of the world. Visualization and modeling are technologies that enable such advancement. As spatiotemporal data acquisition grows exponentially at a rapid pace, powerful and robust visualization and modeling tools are urgently needed to decipher forms, patterns, and relationships embedded in spatiotemporal data to understand geographic dynamics. CA, ABM, and spatial evolutionary algorithms (SEAs), numerical modeling techniques developed for specific domains (such as groundwater flows) have successfully demonstrated capabilities to model geographic dynamics. However, the massive amount of ever-growing spatiotemporal data challenges both the computation and visualization powers of these tools. High-performance computing (HPC) has been a major driver to enable numerical modeling of atmospheric and fluvial dynamics but is still at the experimental stage for CA, ABM, and SEA. HPC is one important future direction to the research on visualization and modeling of geographic dynamics [15]. Other important research topics include the development of visual analytical and methodological frameworks for geographic dynamics across multiple scales and multiple domains.

Cross References

▶ Exploratory Visualization

Recommended Reading

1. Tobler, W. R.: A computer movie simulating urban growth in the Detroit region. Econ. Geogr. **46**, 234–240 (1970)
2. Harrower, M.: A look at the history and future of animated maps. Cartographica **39**, 33–42 (2004)
3. Dibiase, D., Maceachren, A.M., Krygier, J.B., Reeves, C.: Animation and the role of map design in scientific visualization. Cartogr. Geogr. Inf. Syst. **19**, 201–214 (1992)
4. Slocum, T. A., Montello, D.R., Fuhrmann, S., Hedley, N.R., Blok, C., Jiang, B., Koussoulakou, A.: Cognitive and usability issues in geovisualization. Cartogr. Geogr. Inf. Sci. **28**, 61–75 (2001)
5. Monmonier, M.: Summary graphics for integrated visualization in dynamic cartography. Cartogr. Geogr. Inf. Syst.19, 23–36 (1992)
6. Gahegan, M., Hardisty, F., Takatsuka, M., Wheeler, M.: Introducing Geo VISTA Studio: an integrated suite of visualization and computational methods for exploration and knowledge construction in geography. Comput. Environ. Urban Syst. **26**, 267–292 (2002)
7. Kwan, M.-P.: GIS Methods in time-geographic research: geocomputation and geovisualization of human activity patterns. Geogr. Ann., Ser. B Hum. Geogr. **86**, 267–280 (2004)
8. Bithell, M., Macmillan, W.D.: Escape from the cell: Spatially explicit modeling with and without grids. Ecol. Modelling **200**, 59–78 (2007)
9. Batty, M.: Agents, cells, and cities: new representational models for simulating multiscale urban dynamics. Environ. Planning A **37**, 1373–1394 (2005)
10. Torrens, P. M.: Simulating sprawl. Ann. Assoc. Am. Geogr. **96**, 248–275 (2006)
11. McIntosh, J., Yuan, M.: Assessing similarity of geographic processes and events. Trans. GIS **9**, 223–245 (2005)
12. Dietzel, C., Clarke, K.C.: Decreasing computational time of urban cellular automata through model portability. GeoInformatica **10**, 197–211 (2006)
12. Miller, H. J.: Measuring space-time accessibility benefits within transportation networks: basic theory and computational procedures. Geogr. Anal. **31**, 187–212 (1999)
14. Komarov, A. S., Palenova, M.M., Smirnova, O.V.: The concept of discrete description of plant ontogenesis and cellular automata models of plant populations. Ecol. Modelling **170**, 427–439 (2003)
15. Armstrong, M.P., Cowles, M.K., Wang, S.: Using a computational grid for geographic information analysis: a reconnaissance. Prof. Geogr. **57**, 365–375 (2005)

Geographic Information

▶ Market and Infrastructure for Spatial Data

Geographic Information Retrieval

▶ Retrieval Algorithms, Spatial

Geographic Information Sciences

▶ Information Services, Geography

Geographic Information Systems

► Information Services, Geography
► Spatial Decision Support System

Geographic Knowledge Discovery

HARVEY J. MILLER
Department of Geography, University of Utah,
Salt Lake City, UT, USA

Synonyms

Spatial data mining; Geographic data mining; Exploratory spatial analysis; Pattern recognition in spatial data; Hypothesis validation in spatial data

Definition

Geographic knowledge discovery (GKD) is the human-centered process of using computational methods and visualization to explore massive geo-referenced digital databases for novel and useful geographic knowledge. GKD is an extension of the more commonly known process of knowledge discovery from databases (KDD), with "data mining" being a central activity of this process.

Main Text

Geographic information science exists in an increasingly data-rich environment facilitated by the development of high spatial and temporal resolution environmental monitoring devices, location-aware technologies, information infrastructure for data sharing and interoperability, as well as reductions in data storage costs. Traditional spatial analytical methods were developed when data collection was expensive and computational power was weak. The increasing volume and diverse nature of digital geographic data easily overwhelm techniques that are designed to tease information from small, scientifically sampled and homogeneous datasets. They are also confirmatory and require the researcher to have *a priori* hypotheses, meaning that they cannot discover unexpected or surprising information.

Geographic knowledge discovery (GKD) attempts to leverage the continuing exponential growth in computing power to find novel and useful geographic knowledge hidden in the unprecedented amount and scope of digital geo-referenced data being collected, archived and shared by researchers, public agencies and the private sector. This is a human-centered process that involves activities such as developing background knowledge as a guide, selecting, cleaning and reducing the data to make it usable, data mining through the application of efficient, scalable techniques for extracting patterns, interpreting these patterns and constructing knowledge. These stages are not necessarily sequential and are often applied in an iterative manner.

Cross References

► Time Geography

Recommended Reading

1. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery: An overview. In: Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Ulthurusamy, R. (eds.) Advances in Knowledge Discovery and Data Mining, pp. 1–34. MIT Press, Cambridge, MA (1996)
2. Gahegan, M.: On the application of inductive machine learning tools to geographical analysis. Geogr. Anal. **32**, 113–139 (2000)
3. Miller, H.J., Han, J.: Geographic data mining and knowledge discovery: An overview. In: Miller, H.J., Han, J. (eds.) Geographic Data Mining and Knowledge Discovery, pp. 3–32. Taylor and Francis, London (2001)

Geographic Market Segmentation

► Geodemographic Segmentation

Geographic Markup Language

► Web Feature Service (WFS)

Geographic Metadata

► Metadata

Geographic Phenomena

► Geographic Coverage Standards and Services

Geographic Profiling

► CrimeStat: A Spatial Statistical Program for the Analysis of Crime Incidents

Geographic Resources Analysis Support Software

► GRASS

Geographic Spatial Regression

▶ Spatial and Geographically Weighted Regression

Geographic Weighted Regression (GWR)

▶ Spatial Regression Models

Geographical Analysis

▶ Crime Mapping and Analysis
▶ Data Analysis, Spatial

Geographical Information Retrieval

▶ Internet-Based Spatial Information Retrieval

Geographically Weighted Regression

▶ Data Analysis, Spatial

Geography Markup Language (GML)

RAIMUNDO F. DOS SANTOS JR., CHANG-TIEN LU
Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA

Synonyms

GML; Format; Open source; Interoperability; Extensibility; Exchange format; OGC; ISO; Resource description framework (RDF); Properties, simple; Properties, geometric; Reference system, temporal; Reference system, spatial; Features; Schema

Definition

Geography Markup Language (GML) is an open-source encoding based on the eXtensible Markup Language (XML), and suitable for the representation of geographical objects. Organized as a hierarchy of features, collections, and geometries, among other structures, GML objects are modeled after real-world entities characterized by properties and state. In addition, GML has been defined as an information exchange and storage format with which disparate systems can share common geographic data. GML schemas establish a standard blueprint of how geographic objects can be defined by one system and understood by others in a vendor-independent manner.

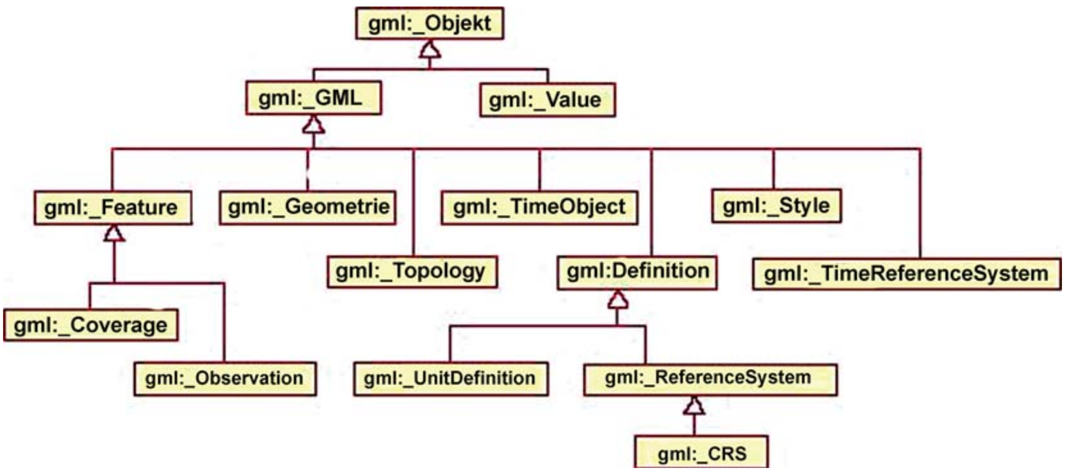
Historical Background

Advances in geospatial applications have promoted the creation of technologies more suitable to handle geographic data. As one of those technologies, GML was developed and released by the Open Geospatial Consortium (OGC), an international entity formed by various public and private organizations. The OGC adopted XML as defined by the World Wide Web Consortium (W3C) as its foundation for the development of GML. By obtaining acceptance from industry, government, and academia, the GML specification has been incorporated as a document of the International Organization for Standardization (ISO).

The development of **GML 1.0** began in 1999. In its first version, it used the *Resource Description Framework* (RDF) metadata model to make statements about resources. Known as triples, RDF statements combine the concepts of subject, predicate, and object to describe a particular object and its characteristics. GML 1.0 defines *Simple Properties* such as strings, numbers, and Boolean values, and *Geometric Properties* such as points and lines. GML 1.0 was a major initiative in the geospatial domain, but more importantly, it served as a stepping stone towards the development of a more robust language.

Under **GML 2.0**, released in 2001, the OGC introduced GML Schemas as a more flexible approach for describing and validating geographic objects. While the language itself was enhanced with more sophisticated structures, the OGC maintained its goal of easy interoperability and extensibility. Three base schemas, namely Geometry, Feature, and XLink, were released with the goal of supporting geospatial applications for specific domains. In addition, it provided a framework for better utilization of linked features in distributed data sets. GML 2.0 addresses mostly two-dimensional simple features, and introduced better support for namespaces to distinguish among features and properties defined in different application domains.

With the advent of **GML 3.0** in 2003, the OGC has made significant additions to the number and capabilities of its schemas. As real-world phenomena is not just limited to the one and two-dimensional spaces, the support for features with complex, non-linear, 3D geometry has been added to the language. Other enhancements include the support of 2D topology and the definition of coverages as subtypes of features. For applications in which historical data and localized timestamps are important, GML 3.0 provides temporal references systems. Units of measurement and default styles for visualization have also been



Geography Markup Language (GML), Figure 1 A Subset of the GML Class Structure [3]

```
<gml:location>
  <gml:Point gml:id="xy1"><element ref="gml:boundedBy"/>
    <gml:pos>-21.4 122.993</gml:pos>
    <gml:LocationString>City of Reston</gml:LocationString>
  </gml:Point>
</gml:location>
```

Geography Markup Language (GML), Example 1 Simple GML Features

defined in this specification. As of this writing, version 3 is the current release.

Scientific Fundamentals

System designers can extend the GML base schemas to create an Application Schema suitable to their specific application domains. While many schemas comprise the GML Specification, a single application normally would only implement a subset of them. For example, Feature is one of the main components of GML, and therefore most applications import the *feature.xsd* schema. The system designer, however, need only import the *topology.xsd* schema if the features in question store topological characteristics. Under GML 3, components are defined according to the Class Hierarchy of Fig. 1 [3]:
The hierarchical structure of Fig. 1 illustrates the parent-child relationship that binds GML components. Each class represents one or more aspects of a real-world entity, helping describe its features, geometric characteristics, location, time of existence, spatial reference system, and many other qualities. Within the various application schemas, Designers may use components "as is" or they may choose to restrict or extend the GML base types to tailor the language to their own needs. "*gml:_Object*" is the abstract class that represents a GML data set or document. Two important components of the GML Specification are the *Feature* and *Feature Collection* elements, represented in the diagram by the *GML:_Feature* class. A feature (or a collection of features) can be a road, a river, a building or any geographic entity that needs representation. Features may have several properties defined, such as *gml:location, gml:boundedBy*, and *gml:LocationString*, as in the code fragment (Example 1):
The *gml:Geometry* abstract class defines objects that can be represented as *points, curves*, and *polygons*. The middle of a region may be described as a point having a *centerOf* property set to a given value. An element may be identified by its *id*, a name, and a description. It may be associated with a spatial reference system (attribute "*gml:SRSReferenceGroup*"). Some of the geometry types currently supported are *gml:Point, gml:Curve, gml:LineString, gml:_Surface, gml:Polygon, gml:_Ring, gml:Circle*, and *gml:LinearRing*. Geometries are often qualified with the Coordinate Reference System to which they belong. In Example 2, the *gml:coordinates* element describes a point in the WGN84 coordinate system having $x = 22.21$ and $y = 31.99$. The point itself may represent an

```
<gml:Point gml:id="gr065" srsName="urn:ogc:def:crs:WGN84">
  <gml:coordinates>22.21, 31.99</gml:coordinates>
</gml:Point>
```

Geography Markup Language (GML), Example 2 Coordinate Reference Systems

```
<gml:TimePeriod gml:id="period01">
  <gml:begin>
    <gml:TimeInstant gml:id="instant01">
      <gml:timePosition>2006-12-04</gml:timePosition>
    </gml:TimeInstant>
</gml:TimePeriod>
```

Geography Markup Language (GML), Example 3 GML Time Elements

```
<gml:FeatureStyle featureType="exp:Town">
  <gml:GeometryStyle>
    <gml:style>fill:yellow</gml:style>
  </gml:GeometryStyle>
</gml:FeatureStyle>
```

Geography Markup Language (GML), Example 4 Styling Components

island in the ocean, a building in the city or just a point on the ground.

Temporal properties are some of the most recent additions to the GML 3 Specification. Features may be associated to time elements with the use of such properties as *gml:timePosition, gml:timePeriod, and timeInstant* (Example 3).

To describe an event, an action that occurs over a period of time, GML provides the abstract *TimeSlice* element. According to the GML Specification, a time slice describes a process-oriented event, such as object changes over time, whereas a timestamp relates to a specific snapshot of the object. The *gml:MovingObjectStatus* element is able to capture the conditions that a moving object experiences over an elapsed period. The *history* and *track* elements may also be used to observe the lifecycle of an object. The components described above do not represent an exhaustive list of time-specific components.

GML makes a strict separation between data and presentation. None of the GML constructs have a native capability of presenting data according to a styling method. For this reason, a default styling mechanism can be used in conjunction with a GML data set to provide proper styling. The term "default" specifies that the styling mechanism that comes with a data set may be used or ignored altogether. The *Feature* class defines certain components useful in the visualization process, such as *GraphStyle, LabelStyle, GeometryStyle*, and *TopologyStyle* objects as exemplified in Example 4:

Querying is one of the most important functions in information retrieval. A GML query language must be flexible enough to support querying and retrieving both spatial and non-spatial data. There have been some attempts at the creation of query languages for GML data. However, most are native to XML, and not necessarily designed for GML. Some examples are XML Query Language (XQL), XML-QL, Quilt, XQuery, and Lorel [2]. XQuery [1], a W3C standard, has been one of the most popular approaches for XML querying, and for GML querying by association [4]. Querying on GML data can be made more efficient when the querying language is able to handle spatial constructs. For example, regions defined by a *Minimum Bounding Rectangles* (MBR) can be manipulated more efficiently with *overlaps, contains, intersects*, and other spatial operators. Similarly, spatial joins are a powerful way of aggregating and linking spatial features that are common in GML data sets.

Key Applications

GML has found its way in many fields of application including, but not limited to, Planning and Resource Management, Mapping and Cartography on Mobile Devices, Real-Time Emergency Response Services, and Image Processing in Data Streams.

Planning and Resource Management

GML is becoming the *de facto* standard for the representation, exchange, and storage of geographic data. Local, state, and federal government agencies have increasingly adopted the use of GML for real estate zoning, area mapping and planning, insurance determination, vegetation monitoring, and coastal erosion. For example, GML applications can be constructed to map and visualize different views of the city, specific sections prone to flooding, sources of pollution, and historical time slices of water bodies. Even though spatial data can be voluminous, GML allows users to work with a small subset of the total data that more closely relates to their interest.

Mapping and Cartography on Mobile Devices

Advances in wireless networks have contributed to the widespread use of mobile devices for both business and personal purposes. GML has the potential to fill the gap for mapping applications in PDAs and cell phones where location-based services use the constant exchange of data from GPS systems to synchronize moving objects with remote servers and perform nearest-neighbor searches.

Real-Time Emergency Response Services

GML is modularized in such a way that the entire data set or just a small subset can be analyzed at a time. By managing Minimum Bounding Rectangles (MBR), users can search and view only the information that pertains to them. For example, highways in the North part of the city can be monitored for unusual activity where fog is prominent during certain hours of the morning. By gathering relevant information and filtering out unimportant features, emergency personnel can focus their attention on a smaller area and respond as needed on a near real-time basis.

Image Processing in Data Streams

GML is designed to support streaming data flows. As such, it provides robust facilities that can handle incoming information from various remote sources. This is especially helpful in dynamic environments where changes occur frequently. Applications for satellite imagery and remote sensing are typical examples. Weather data, for instance, are commonly generated from satellite feeds with large volumes of information in short time intervals. Meteorological agencies rely on the frequent and correct processing of this information for fast and accurate forecasts.

RSS News Feeds

In the recent past, GML has been proposed as a data delivery medium to be used in conjunction with RSS news feeds. The idea has seen considerable attention as the RSS format is boosted by the popularity of blogs and internet news sites. GML geometries along with other components such as temporal constructs may be leveraged to make application data more relevant in specific contexts. Political sites, for example, may want to describe the president's trip according to visited locations, spatial characteristics of the areas in question, and other annotations that support the evolving story. While an official schema has not been published as of this writing, this area of application promises to make geospatial data more readily available not only to members of the geospatial community, but also to casual readers of internet news.

Future Directions

Even though GML has gained increased acceptance, there is still much work that can be done to improve its usefulness and efficiency. GML may benefit from a query language that supports spatial queries such as overlaps, intersections, ranges, and nearest neighbors. The objects represented in GML are often more complex than those typically encoded in XML, since geographic objects have both spatial and non-spatial attributes. The data model for a GML query language has to reflect this complexity. Therefore, current XML query languages must be extended to support the rich array of elements that GML makes available. GML data needs indexing just as much as other spatial data sources. Because it is encoded in semi-structured XML, GML data must be parsed out with tools such as SAX (Simple API for XML) or DOM (Document Object Model). The extracted data in turn can be indexed using one of several approaches such as R Trees or Quad Trees. The exact approach most suitable for GML needs further evaluation. Integrating GML with mobile devices is a promising direction. Mobile environments are inherently challenging due to their constrained nature. Adapting GML for low-bandwidth networks and limited-storage devices will help bring those devices inline with other distributed data sources that hold important geospatial information.

Cross References

► Data Models in Commercial GIS Systems
► Geospatial Semantic Web
► Indexing, Hilbert R-tree, Spatial Indexing, Multimedia Indexing

Recommended Reading

1. XQuery 1.0: An XML Query Language, W3C Working Draft. http://www.w3.org/TR/xquery/. Accessed 3 Dec 2006
2. Bonifati, A., Ceri, S.: Comparative Analysis of Five XML Query Languages, ACM SIGMOD Record, vol. 29, pp. 68–79, March, (2000)
3. Cox, S., Daisey, P., Lake, R., Portele, C., Whiteside, A.: OpenGIS Geography Markup Language (GML 3.0) Implementation Specification, OpenGIS Specifications. http://www.opengis.org/specs/?page=specs. Accessed 3 Dec 2006
4. Li, Q., Moon, B.: Indexing and Querying XML Data for Regular Path Expressions. In: Proceedings of the 27th International Conference on Very Large Data Bases, Rome, Italy, pp. 361–370 (2001)
5. Peng, Z.R., Tsou, M.H.: Internet GIS – Distributed Geographic Information Services for the Internet and Wireless Networks. Wiley Publishing, Hoboken, New Jersey (2003)
6. Shekhar, S., Chawla, S.: Spatial Databases: A Tour. Pearson Education, Inc., Upper Saddle River, New Jersey (2003)

Geoinformatic Surveillance

► Hotspot Detection, Prioritization, and Security

GEOINT

► Intelligence, Geospatial

Geo-Intelligence

► Intelligence, Geospatial

Geolocation

► Indoor Positioning, Bayesian Methods

Geo-Mashups

► Computer Environments for GIS and CAD

Geomedia

► Intergraph: Real Time Operational Geospatial Applications

Geometric Engine Open Source (GEOS)

► Open-Source GIS Libraries

Geometric Fidelity

► Positional Accuracy Improvement (PAI)

Geometric Modeling

► Vector Data

GeoOntologies

► Knowledge Representation, Spatial

Geo-Portal

► deegree Free Software

Geopriv Group, IETF

► Privacy Threats in Location-Based Services

Georectified

► Photogrammetric Products

Georegistration

► Registration

Geo-Role-Based Access Control

► Security Models, Geospatial

GEOS Library

► PostGIS

Geosensor Networks

Silvia Nittel
Department of Spatial Information Science
and Engineering, University of Maine, Orono, ME, USA

Synonyms

Sensor networks; Environmental sensor networks; GSN

Definition

Advances in small, low-cost microelectronic and mechanical systems (MEMS) with limited on-board processing and wireless communication capabilities, and the development of novel sensor materials enables us to build a new generation of technology that consists of large collections of untethered, small-form, battery-powered computing nodes with various sensing functions. These sensor nodes can be densely distributed over a geographic region, and are able to measure environmental processes, such as weather development, seismic activity, or track the movements of toxic fumes at a level of detail that was not possible before. The continued trend towards miniaturization and inexpensiveness of sensor nodes makes it possible that such sensor nodes are less than a cubic millimeter in size, and sensor networks can be made up of thousands or even millions of sensors.

G*eosensor networks* (GSN) are a specialized application of sensor network technology to monitor, observe and track environmental phenomena and processes [11]. Geosensor networks are deployed in geographic space to monitor environmental phenomena at a high spatio-temporal scale and resolution, and the data is available in near real-time. For example, a sensor network was deployed on Great Duck Island in Maine to observe the nesting behavior of an endangered species of storm petrels [1]. Another sensor network consisting of 60 motes was deployed to monitor the microclimate around a single Redwood tree over a period of several months [2].

Historical Background

From telescopes to microscopes, humans have developed instruments to monitor and observe the world in ways that are not obvious to the human eye and senses. Many technological innovations have been made over the past century to sense information within the environment. Some of the most impressive and powerful instruments are, e. g., remote sensing satellites deployed in space. Such instruments enable scientists to monitor processes on the earth at a global scale. With local weather stations, rainfall samples and/or wind direction and speed can be measured. Recording such information allows the analysis of long-term trends in climate development on an entire continent. Today, it is characteristic that sensors are stationary, expensive and thus, sparsely distributed over a large geographic area. Data is logged and stored locally, and often retrieved in batch mode manually or via satellite link. Thus, dynamic complex processes at a small and local scale are more difficult to observe due to the scale of the existing instruments. During the mid-1990s, the miniaturization of computing devices and the development of novel sensor materials and microsensors lead to the technology of wireless sensor networks. Sensor networks consist of a large number of small computing devices with attached sensor boards, and equipped with batteries and wireless communication. They can be deployed in a local environment in an untethered way, and sense the environment at temporally and spatially high-resolution in real-time.

Sensor networks, however, have the following constraints that pose novel challenges from a system and application developmental standpoint:

Power consumption Sensor nodes are limited with regard to their battery supply, and energy conservation is a major system design principle.

Low-range communication The bandwidth and range of wireless communication is limited. Since communication is a much higher drain on the energy consumption than on-board processing, optimizing communication within the sensor network is a major system design consideration.

Limited computing and storage capabilities Sensor nodes have, at least for the foreseeable future, limited on-board computational, volatile, and persistent storage capabilities. Thus, on-board data processing using available memory and CPU is also limited.

Self-organization Due to the large number of sensor nodes, the failure rates of nodes, and the often unattended deployment, task management and handling in sensor networks is decentralized and self-organizing. Thus, some level of local autonomy must be provided for the devices.

Scientific Fundamentals

The deployment of sensor networks provides outstanding ability to monitor discrete and continuous phenomena in physical space.

Over the last years, much research has focused on the design of platforms, programming languages for such constrained environments (e. g., nesC), robust, energy-efficient communication protocols [3], and the development of sensor materials and sensor devices. A major task of sensor networks, however, is their data collection and processing instrumentation. Albeit their powerful and novel capabilities to observe the physical world, today programming sensor networks is cumbersome due to the failure-prone nature of nodes and communication links, and the vast parallel computational nature of such systems.

A user, often a domain scientist, needs to define the necessary tasks in a user-friendly way, and delegate the optimization and ultimately self-adaptive execution to the run-time environment without having to worry about the details. From a database perspective, a sensor network can be viewed as a distributed database system (DBS) with sensor nodes that run lightweight versions of the DBMS themselves. The DBMS supports a single sensor table and the attributes match the attached sensors such as temperature, humidity, location, etc. The query syntax is similar to SQL-style queries and is extended with sampling epochs for continuous queries. Thus, a user can interact with the sensor network as a virtual single sensor, and send declarative queries to it. Each sensor node runs a tiny footprint DBS locally, and participates in global distributed query execution. Tuples for the sensor table are created in an append-only style created by all sensor nodes in the sensor network. The first sensor database system prototypes are TinyDB [4] and Cougar [5].

Typical sensor network queries are spatio-temporal. For example, a user might be interested in hourly measurements of temperature values along the Redwood tree to observe the local microclimate around the tree:

```
SELECT sensor.id, temperature FROM sensors
SAMPLE PERIOD 60min.
```

Take note that queries define the requested data in a declarative way, and the user does not need to know the availability or node identifier of sensor nodes, nor does he/she deal with the details of the execution of the query. The DBMS distributes the query to the sensor nodes of interest using a geo-routing or a tree-based routing protocol. Similarly, query results are accumulated and routed back to the base station using a tree-based protocol. Using an energy-efficient data aggregation tree, nodes combine local partial data aggregation results with routing partial results back to the query originating sensor node.

Currently, spatial queries over a sensor network often retrieve discrete information measured at the location of sensor nodes. Many environmental phenomena, such as a temperature field or a gas concentration field in an open space are spatio-temporally *continuous*. Individual sensor readings, however, are point samples taken at the physical location of the sensor nodes about the underlying phenomenon [5,6]. For geosensor networks, the challenge exists to provide an accurate and precise estimation of all points of dynamic spatial field based on limited discrete point samples collected within a geosensor network. Since sensor nodes are energy- and processing-limited due to their small-form factor and battery power, estimation algorithms have to be lightweight and processed in a distributed manner 'in the network'.

Another setting for geosensor networks are sensor nodes that are *mobile*, such as sensor nodes installed within automobiles or used by pedestrians, attached to animals, floating on the ocean [10] or embedded in the road pavement or traffic lights. Here, up-to-date information is collected and exchanged in an ad-hoc, peer-oriented fashion between nodes in spatial proximity [9]. Neighboring sensor nodes in intelligent transportation systems can inform other existing nodes in the same region about environmental events such as icy patches or other unexpected road hazards that can be sensed on the road.

Key Applications

The application areas for geosensor networks are plentiful and can be found in the following areas: biodiversity, biogeochemical cycles, climate change, infectious diseases, invasive species, carbon cycle, earthquakes and tsunami warning systems, coral reefs observation, animal and habitat monitoring, and coastal and ocean observation. Some applications have been mentioned before [1,2].

It is typical for geosensor networks that the technology is integrated with existing, larger-scale sensing platforms such as remote sensing instruments, buoys, autonomous underwater vehicles, wind mills, or weather stations to integrate sensed data for the same spatial region at various spatial and temporal scale. When combining sensor networks with existing sensing platforms, so-called *sensor webs* are created. Similar to the idea of the World Wide Web, the vision exists that sensors and sensor networks should be accessible and usable in a uniform way so that scientists can find, combine, and query real-time sensors in a geographic region for a specific application. This leads to the fact that (often expensive) sensor platforms are more reusable for different purposes. The OpenGIS Consortium provides a standardization approach for sensor platform interfaces and data exchange (Sensor Web Specifications) for sensor webs. Today, several research networks have been established in the area NEON (National Ecological Observatory Network), GLEON (Global Lake Ecological Observatory Network), CREON (Coral Reef Environmental Observatory Network), and the NEPTUNE Cyberinfrastructure.

Future Directions

Currently, the actual application and deployment of geosensor networks is in its infancy. With the increasing robustness of sensors, wireless communication, improved battery life and technology, and software developments, geosensor networks will be a powerful, added technology to the existing environmental sensing platforms [12].

Cross References

► Geosensor Networks, Estimating Continuous Phenonena
► Geosensor Networks, Qualitative Monitoring of Dynamic Fields

Recommended Reading

1. Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D.: Wireless sensor networks for habitat monitoring In: 1st Int'l Workshop on Wireless Sensor Networks and Applications, Atlanta, Georgia 2002
2. Tolle, G.,Polastre, J., Szewczyk, R., Culler, D., Turner, N., Tu, K., Burgess, S., Dawson, T., Buonadonna, P., Gay, D., Hong, W.: 2005. A Macroscope in the Redwoods. International Conference on Embedded Networked Sensor Systems (ACMSensys), San Diego, November (2005)
3. Woo, A., Tong, T., Culler, D.: Taming the underlying challenges of reliable multihop routing in sensor networks. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (ACMSensys), pp. 14–27. Los Angeles, November 2003
4. Madden, S., Franklin, M.J., Hellerstein, J., Hong, W.: TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks. In: Proceedings of the 5th USENIX Symposium on Operating Systems Design and Implementation (OSDI), pp. 131–146. December 2002
5. Fung, W.F., Sun, D., and J. Gehrke, J.: COUGAR: The Network is the Database. In: Proceedings of International ACM SIG Conference Modeling of Data (SIGMOD), Madison, WI 2002
6. Sharifzadeh, M., Shahabi, C.: Supporting spatial aggregation in sensor network databases. In: Proceedings of ACM Symposium Advanced in Geographic Information Systems (ACM-GIS), pp. 166–175. Washington, DC (2004)
7. Hellerstein, J.M., Hong, W., Madden, S., Stanek, K.: Beyond Average: Towards Sophisticated Sensing with Queries. In: Proceedings of Workshop on Information Processing In Sensor Networks (IPSN), 2003
8. Jin, G., Nittel, S.: NED: Efficient Event Detection in Sensor Network. In: Proceedings of MDM-Workshop "Mobile Location-Aware Sensor Networks", Nara, Japan 2006
9. Nittel, S., Duckham, M., Kulik, L.: Information Dissemination in Mobile Ad-hoc Geosensor Networks. In: Proceedings of the 3rd International Conference on Geographic Information Science (GIScience 2004), Springer LNCS, College Park, Maryland 2004
10. Nittel, S., Trigoni, N., Ferentinos, D., Neville, F., Nural, A., Pettigrew, N.: A drift-tolerant model for data management in ocean sensor networks In: Proceedings of MobiDE'07, in conjunction with SIGMOD, Beijing, China 2007
11. Stefanidis, A., Nittel, S.: Geosensor Networks, CRC Press, Boca Raton (2004)
12. Nittel, S., Labrinidis, A., Stefanidis, A.: Advanced in Geosensor Networks. Springer LNCS (in press 2007)

Geosensor Networks, Estimating Continuous Phenonena

GUANG JIN
Department of Spatial Information Science and Engineering, University of Maine, Orono, ME, USA

Synonyms

Interpolation of continous geofields; Estimation predication; Phenomenon spatial field; Sample trial; Sensor networks; Energy constraint; Estimation, parametric; Estimation, non-parametric; Tiny aggreagation service; Distributed algorithm

Definition

Geosensor networks (GSN) are deployed to monitor different environmental phenomena over a spatiotemporal space. Many environmental phenomena (also called *spatial fields*), such as a temperature field or a gas concentration field in an open space, are spatiotemporally continuous within the spatiotemporal space. Individual sensor readings, however, are point samples taken at the physical location of the sensor nodes about the underlying phenomenon. Thus, neighboring sensor nodes' readings are likely similar.

For GSN, the challenge is to provide an accurate and precise estimation of all points of dynamic spatial field based on limited discrete point samples collected by the network. Since sensor nodes are energy and processing limited due to their small-form factor and battery power, estimation algorithms have to be lightweight and processed in a distributed manner "in the network".

Main Text

Sensor database management systems (SDMS) provide users an intuitive and easy-to-learn programming interface to define observation tasks for phenomena, and can also serve to estimate and track continuous phenomena. One of the most challenging objectives of SDMS is to provide accurate and precise estimation results for continuous phenomena based on limited point samples processed in real-time within the network in a distributed and collaborative way. Since GSN are energy- and computing-constrained, studies on estimating continuous phenomena in GSN have to maintain a graceful balance between the quality of the estimation results and the resource consumption within GSN.

Traditionally, spatiotemporal point samples are collected in advance and processed on a centralized computer. Generally speaking, there are two types of methods on how to estimate continuous phenomena from point samples.

Parametric estimation methods estimate an underlying phenomenon as a function. The function is usually a linear combination of a set of basic functions. For example, in a polynomial regression, $f(x, y) = \beta_0 + \beta_1(x) + \beta_2(y) + \beta_3(xy) + \varepsilon$, different ordered polynomials are the basic functions. The parameters, βs, are chosen to minimize the error, ε.

Nonparametric estimation methods require no prior specified model structure, and estimate an underlying phenomenon directly from the samples. For example, moving average is a simple nonparametric estimation, which estimates the phenomenon reading as the average sample value from a moving region.

Wireless communication between sensor nodes is very expensive, in terms of energy consumption from the battery-driven sensor nodes. A traditional centralized approach to estimating the dynamic field of a continuous phenomenon, however, requires large amounts of wireless communication to transmit raw readings from the distributed nodes to a central base computer. An efficient estimation solution in GSN needs to use distributed algorithms that are executed within the network to minimize the communication cost. Two types of communication cost, the processing and the result representation cost, are necessary for any estimation method in GSN. These two communication costs are also important to evaluating and comparing the efficiency of different algorithms.

In many cases, a global average is useful to estimate a phenomenon. Tiny AGgregation (TAG) service [1] presents an in-network solution to computing an average of sensor readings using a routing tree structure [2]. In TAG, a node computes the average result based on its own reading and its children's readings. The node sends its local partial result to its parent. Finally, the root is able to compute the global average based on all its children nodes. TAG requires only a concise communication package to represent a partial average, and processes the partial results locally. However, if the sensor nodes are not evenly distributed over a monitoring region, the simple average usually returns faulty results. A weighted spatial average [3] applies a weight to each sensor reading, and returns more accurate results. For example, a monitoring region can be partitioned into a Voronoi diagram based on the sensor nodes' location. The reading for a Voronoi cell can be weighted by the size of cell, and aggregated along the routing tree. Generating the Voronoi diagram, however, requires additional processing costs.

Often, users prefer a more fine-grained estimation result than a simple average value over an entire region or a subregion. TAG has been extended [4] to return a fine-grained estimation. For example, each sensor node can represent a grid cell in a monitoring region. A node can fill its reading to the corresponding grid. Nearby similar readings can be represented by a larger polygon region. In this way, the root node can generate a contour map to estimate the underlying phenomenon. However, representing the contour map requires additional costs and has to be simplified to save expensive wireless communication [5]. Parametric estimation methods, such as kernel regression [6], have also been applied to estimate underlying phenomena in GSN. A set of kernel functions are used as the basic functions. In this way, only the estimated values of kernel functions are necessary to represent the underlying phenomena. However, to compute the optimal parameter values, complex matrix operations are required, which consume large amounts of communication in GSN.

The constraints of GSN, especially the expensive wireless communication, challenge efficient phenomenon estimation methods in GSN. An efficient estimation solution has to return high quality results while still preserving the processing and representing requirement.

Cross References

► Geosensor Networks
► Geosensor Networks, Qualitative Monitoring of Dynamic Fields

Recommended Reading

1. Madden, S., Franklin, M.J., Hellerstein, J.M., Hong, W.: TAG: a Tiny AGgregation service for ad-hoc sensor networks. SIGOPS Operating Systems Review, 36, pp. 131–146. ACM Press, New York (2002)
2. Woo, A., Tong, T., Culler, D.: Taming the underlying challenges of reliable multihop routing in sensor networks. In: Proceedings of the First International Conference on Embedded networked sensor systems, California, USA, 4–7 Nov 2003, pp. 14–27. ACM Press, New York (2003)
3. Sharifzadeh, M., Shahabi, C.: Supporting spatial aggregation in sensor network databases. In: Proceedings of the 12th annual ACM international workshop on Geographic information systems, Washington DC, USA, 12–13 Nov 2004, pp. 166–17. ACM Press, New York (2004)
4. Hellerstein, J.M., Hong, W., Madden, S., Stanek, K.: Beyond average: Towards Sophisticated Sensing with Queries. In: Proceedings of the Second International Workshop on Information Processing in Sensor Networks, pp. 63–79, Palo Alto, CA, USA, 22–23 Apr 2003
5. Harrington, B., Huang, Y.: In-network surface simplification for sensor fields. In: Proceedings of the 13th annual ACM international workshop on geographic information systems, pp. 41–50, Bremen, Germany, 4–5 Nov 2005
6. Guestrin, C., Bodik, P., Thibaux, R., Paskin, M.A., Madden, S.: Distributed regression: an efficient framework for modeling sensor network data. In: Proceedings of the third international symposium on information processing in sensor networks, Berkeley, California, USA, 26–27 Apr 2004, pp. 1–10. ACM Press (2004)

Geosensor Networks, Formal Foundations

LARS KULIK
Department of Computer Science and Software Engineering, The University of Melbourne, Melbourne, VIC, Australia

Synonyms

Provable properties; Theoretical analysis; Mathematical theory of geosensor networks

Definition

The geometric nature of geosensor networks (GSNs) has sparked an interest in researching their formal foundations. As the research on formal foundations has just begun, there is no single view of what encompasses a theoretical approach to sensor networks. Typically, a theoretical approach applies techniques from computational geometry, theoretical computer science, or mathematics, in particular (algebraic) topology and graph theory. The goal of a formal analysis is to identify fundamental properties of a sensor network, for example, the total number of sensors required if every point in the sensor field has to be covered by a fixed minimum number of sensors. More generally, a formal analysis studies provable properties of GSNs and algorithms used for GSNs. A formal analysis can compute the computational complexity of an algorithm and decide, if a problem has an efficient solution or if it is intractable.

Historical Background

Current approaches to evaluating algorithms in GSNs are mainly based on extensive simulations or the deployment of test-beds. Extensive simulations can analyze an algorithm under a variety of conditions but can only assess algorithms in terms of their average runtime. Test-beds are ideal to determine the performance of an algorithm under realistic conditions but the results are only applicable to the specific settings of the test-bed. A rigorous algorithmic complexity analysis of a geometric algorithm can not only evaluate the average complexity of an algorithm, but more importantly its worst case complexity. Such a complexity analysis can be vital for the life expectancy of a GSN. For example, a geometric routing algorithm that has a low average energy cost, can have a high worst-case cost in certain situations, which could deplete the batteries of a significant number of nodes and thus lead to the failure of the entire GSN.

Scientific Fundamentals

The monitoring, collection, storage, and processing of geospatial data lies at the core of GSNs [1]. Spatial information is vital for GSNs on two counts: first, in sensor networks it is generally not important which sensor has monitored a geospatial phenomenon, but what type of data and *where* this data is sensed within the network; second, the nodes of a GSN often have the ability to determine their own positions and those of other nodes. Geometric and topological approaches exploit the availability of spatial information to research more efficient algorithms and to discover fundamental characteristics of a GSN.

The features of a GSN are primarily determined by the spatial layout of its deployment and its underlying geography. Its spatial configuration controls the topology of the network, in particular, which nodes are in communication range and, therefore, which routing and data collection algorithms are most effective. The proximity of the nodes influences how well areas are covered by the sensor field and affects the achievable resolution monitoring a geospatial phenomenon. The proximity also determines the spatial correlation of the sensor readings, and in turn decides which and how often sensors need to transmit and provide sensor readings. A formal analysis can identify key parameters that determine the global behavior and properties of a sensor network. Theoretical and algorithmic results can assist in designing, configuring, and administering a GSN, and aid in the selection of appropriate algorithms for tasks such as monitoring, tracking, or routing.

The deployment of a sensor network can be formally represented as a graph: vertices represent the sensor nodes and edges the links between the sensor nodes, i. e., specify whether or not two nodes are in communication range. The edge weights can represent the received signal strength. In a location-aware GSN vertices are not only known by their IDs but also have positions, and their edge weights can reflect the Euclidean distance between two nodes in communication range.

Key Applications

There is a large variety of research areas in GSNs in which formal geometric approaches can be applied. The areas include coverage problems, topology control, data collection and aggregation, routing and dissemination, as well as the discovery of the spatial layout in terms of holes and boundaries. This article surveys these five topics and shows how formal approaches can advance them.

Coverage Problems

Research on coverage problems [2] investigates efficient techniques, where to deploy sensors if a certain data quality is required and how to adapt a sensor network such that each point in the sensor field is covered by a certain number of sensors. Computational geometry provides a large range of techniques that can be used for coverage problems. It is known that in order to guard a room with n corners, at least $n/3$ sensors (guards) are necessary [3]. Similarly, if sensors are randomly placed, then the Voronoi cell of a sensor is given by all points in the sensor field that are closer to this sensor than to any other sensor in the network. Assuming that the sensor closest to a location provides the highest data quality and thus should be responsible monitoring that location, the Voronoi diagram determines the responsibility region of each sensor. Computational geometry provides efficient algorithms to compute the Voronoi cells [4].

Topology Control

The topology of a sensor network is determined by the communication range of each sensor node. As communication is expensive, it is imperative to adjust the communication range to save energy. The key question in topology control [5] is: given a set of sensor nodes, what is the critical transmission range to ensure a connected network? As sensor nodes are cheap devices, most approaches assume that all nodes have the same transmission range. If the locations are precisely known, then controlling the topology of a sensor network can be formally modeled as a minimum spanning tree problem. A minimum spanning tree of a graph is a subgraph, more precisely a tree, with the same vertices that connects all vertices and minimizes the sum over all edge weights. The critical transmission range is the longest edge in the minimum spanning tree. There are a number of algorithms in graph theory to compute the minimum spanning tree efficiently in a distributed manner [6]. The theory of geometric random graphs [7] allows the computation of the critical transmission range even if the node locations are not precisely known, as long as the nodes are randomly and uniformly distributed. Random geometric graphs are generated by randomly uniformly placing points into an area (often the unit square) and connecting any pair of points by an edge if their distance is smaller than a predetermined threshold.

Data Collection and Aggregation

Efficient collection and aggregation algorithms are vital for larger GSNs. A theoretical analysis [8] shows that for any wireless network the throughput per node converges to zero if the number of participating nodes increases in an architecture where each node separately reports to a single processing node. Aggregation algorithms trade communication for computation, as a sensor node consumes significantly less energy for processing than for communication. Instead of transmitting the raw data from each node individually, nodes compute aggregates such as the sum or the average of sensor readings and only transmit those aggregates. As a result, aggregation not only prolongs the life expectancy of a network, but also increases the scalability of the network. The problem of collecting data from a set of sensor nodes in the most efficient way back to a central sink can be regarded as an instance of the minimum Steiner tree problem [9]: given a set of sensor nodes, the Steiner tree is the minimum spanning tree that includes those nodes. This problem is known to be NP-complete (Non-deterministic Polynomial time), which means there is no efficient way known to compute the Steiner tree. This has initiated a variety of local algorithms that can approximate the optimal collection tree [10].

Routing and Dissemination

In routing or dissemination, the task is to route a packet between two or more nodes along an optimal path, which might be the shortest, most reliable, or most energy-efficient path. Geometric routing [11] assumes that a node knows the locations of itself, its neighbors in communication range, and the destination node. Geometric routing protocols are lightweight because they do not need to maintain or discover routes, as is the case for topological routing protocols, which are based on link information only. A criticism often put forward for location awareness is that the energy cost for acquiring coordinates can be high and unreliable. The underlying idea of geometric routing is the use of a local algorithm that forwards a message to a neighbor in communication range that is closer to the destination than itself. A greedy strategy selects among those neighbors in each step the closest one to the destination [12]. Although this greedy strategy is loop-free, it can get stuck if none of its neighbors is closer to the destination than itself. If the set of points that are in communication range of a sensor but are closer to the destination than the sensor itself does not contain any other sensor, then that area is called a void. Since voids can happen frequently, a backup strategy is required. There is a large body of theoretical work on face-routing algorithms [13] that can be used as a fallback strategy. If a graph is planar, it has been shown that face routing along the perimeter of voids can guarantee the delivery of all packets. This is immediately applicable to GSNs as it is possible to compute locally a planar communication subgraph. As location is a powerful concept in routing, current research extends this idea in the form of virtual coordinates to sensor networks that are not location-aware. Virtual coordinates are based on the idea that nearby nodes have similar coordinates, whereas nodes that are far apart have different coordinates. In all sensor networks, nodes have local information about the topology of the network, and the virtual coordinates are computed based on the observed connectivity. Virtual coordinates can be more beneficial as positional information for routing, because nearby nodes might not be connected, for example if they are separated by a wall.

Discovery of the Spatial Layout

Discovering the shape features and spatial layout of a GSN deployment is vital to routing and data collection algorithms. Many algorithms have been evaluated under the assumption that the deployment area of a sensor network is rectangular and the nodes themselves are uniformly distributed. This assumption is often too optimistic, and algorithms that perform well in simulations are less suitable in real GSN deployments. The performance of an algorithm depends on the spatial features of a deployment, in particular its boundary, its medial axis, its communication and deployment holes, its corridors that could bottleneck the transmission of sensor readings, as well as the clustering of the sensor nodes. Algorithms oblivious to those features can prematurely collapse a sensor network. One example: an algorithm that routes information irrespective of the residual energy level always along the boundary of a hole will quickly deplete those nodes and leads to a rapid growth of the hole, which might disconnect the network or exponentially decrease its life expectancy. Furthermore, a dramatic change of the network's boundary could signal a major event such as a fire or a thunderstorm that destroyed significant parts of the network. Algebraic topology [14] and computational geometry both provide techniques to identify shape features of a GSN; however, many techniques assume global knowledge. Research in both areas is currently underway to adapt these techniques for sensor networks.

Future Directions

Although many of the techniques in computational geometry and mathematical topology are often directly applicable to GSNs, they typically assume global knowledge about the sensor network, for example in the computation of a Voronoi diagram or the Steiner tree. However, providing global knowledge at the node level is an exceptionally expensive operation and usually not a viable option in sensor networks. Many algorithms need to be revisited and tailored for local computations that only assume geometric knowledge in the neighborhood of a sensor node. More generally, techniques are sought that can infer from local knowledge of individual nodes, global characteristics of the entire spatial layout of the network, such as its boundary or the number of holes. One approach to inferring global features from local properties is based on combinatorial maps, which can be applied to study spatiotemporal phenomena such as dynamic spatial fields in a qualitative manner [15]. Furthermore, algorithms generally assume that the position of nodes is precisely known. In GSNs, however, node positions are often inexact or not available, and links between nodes can be asymmetric or might be intermittent, posing new challenges for formal approaches to GSNs.

Cross References

► Geosensor Networks, Qualitative Monitoring of Dynamic Fields
► Mereotopology
► Voronoi Diagram

Recommended Reading

1. Stefanidis, A., Nittel, S.: GeoSensor Networks. CRC, Boca Raton, FL (2004)
2. Cardei, M., Wu, J.: Coverage in wireless sensor networks. In: Ilyas, M., Magboub, I. (eds.) Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems, pp. 19:1–19:10. CRC, Boca Raton, FL (2004)
3. O'Rourke, J.: Computational Geometry in C. Cambridge University Press, New York (1988)
4. Berg, M., Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry: Algorithms and Applications. Springer, New York (2000)
5. Santi, P.: Topology control in wireless ad hoc and sensor networks. ACM Comput. Surveys **37**, 164–194 (2005)
6. Gallager, R., Humblet, P., Spira, P.: A distributed algorithm for minimum-weight spanning trees. ACM Trans. Programm. Languages Syst. **5**, 66–77 (1983)
7. Diaz, J., Penrose, M., Petit, J., Serna, M.: Convergence theorems for some layout measures on random lattice and random geometric graphs. Combinatorics Probability Comput. **9**, 489–511 (2000)
8. Gupta, P., Kumar, P.: The capacity of wireless networks. IEEE Trans. Inf. Theory **46**, 388–404 (2000)
9. Krishnamachari, B., Estrin, D., Wicker, S.: The impact of data aggregation in wireless sensor networks. In: Proceedings of the 22nd International Conference on Distributed Computing Systems, Workshops, Vienna, Austria, 2–5 July 2002
10. Kulik, L., Tanin, E., Umer, M.: Efficient data collection and selective queries in sensor networks. In: Nittel, S., Labrinidis, A., Stefanidis, A. (eds.) Advances in Geosensor Networks. Springer, New York (2007, in press)
11. Kuhn, F., Wattenhofer, R., Zhang, Y., Zollinger, A.: Geometric ad-hoc routing: of theory and practice. PODC '03: Proceedings of the 22nd Annual Symposium on Principles of Distributed Computing, Boston, MA, 13–16 July 2003
12. Karp, B., Kung, H.: GPSR: greedy perimeter stateless routing for wireless networks. In: Proceedings of the 6th International Conference on Mobile Computing and Networking, Boston, MA, 6–11 Aug 2000
13. Bose, P., Morin, P., Stojmenovic, I., Urrutia, J.: Routing with guaranteed delivery in ad hoc wireless networks. Wireless Networks **7**, 609–616 (2001)
14. Ghrist, R., Muhammad, A.: Coverage and hole-detection in sensor networks via homology. In: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, Los Angeles, CA, 25–27 Apr 2005
15. Duckham, M., Nittel, S., Worboys, M.: Monitoring dynamic spatial fields using responsive geosensor networks. In: GIS '05: Proceedings of the 13th Annual ACM International Workshop on Geographic Information Systems, Bremen, Germany, 4–5 Nov 2005

Geosensor Networks, Qualitative Monitoring of Dynamic Fields

MATT DUCKHAM
Department of Geomatics, University of Melbourne, Melbourne, VIC, Australia

Synonyms

Ambient spatial intelligence; Qualitative spatial reasoning; Discretization of quatitative attributes; Combinatorial map; Qualitative spatial representations

Definition

Environmental phenomena that vary continuously across regions of space and periods of time, such as changing sea temperature, concentrations of gas pollutant in the air, or levels of soil moisture, are called dynamic spatial fields. Information about dynamic spatial fields is important to a wide range of environmental applications. One of the goals of using a geosensor network (GSN) is to enable improved, more detailed monitoring of dynamic spatial fields. Individual sensor nodes in a GSN usually generate *quantitative* information. For example, a node might record a temperature of 23°C located at coordinates 18.04S, 146.49E at time 20:51:06 on January 7th 2007. However, the information needed by decision makers is typically *qualitative*, and concerns the relationships between groups of spatially and temporally nearby records. For example, an environmental manager may be interested in the whether a high temperature "hotspot" has grown or moved.

Generating qualitative information about dynamic spatial fields within a GSN presents a number of challenges. The most important challenge is to achieve qualitative monitoring using only on local communication between nearby nodes. Resource limitations in GSN mean that global communication, where any node can communicate with any other, is not scalable. Thus, studies of qualitative monitoring of dynamic spatial fields usually assume that at any time an individual node in the network does not have access to global knowledge about the state of the entire network, only to local knowledge about the state of its immediate neighbors.

Historical Background

Qualitative spatial reasoning is concerned with discrete, non-numerical properties of space. There are three main reasons for being interested in the qualitative (as opposed to the quantitative) aspects of geographic space [5]:

- Qualitative properties form a small, discrete domain; quantitative properties form a large, continuous domain, often modeled by real numbers. For example, temperatures in degrees Kelvin are modeled using the set of non-negative real numbers. Yet for some applications, temperature may be adequately modeled as an element from the set {hot, cold, warm}.
- Qualitative properties are *supervenient* on, and derivable from, quantitative properties. For example, in a particular application the numerical temperature 35°C may be described qualitatively as "hot."
- The boundaries between qualities normally correspond to salient discontinuities in human conceptualization of quantitative properties. For example, in coral reef monitoring applications, the qualitative boundary between "warm" and "hot" may be set to correspond to the quantitative temperature at which coral reefs are in danger of coral bleaching.

The literature contains many studies of different qualitative aspects of space, including relative distance [6,9] and direction [3,4], and in particular topological relationships between spatial entities [2,7].

Scientific Fundamentals

With respect to GSNs, the three general reasons for being interested in qualitative aspects of geographic space lead directly to three potential advantages of using qualitative monitoring of dynamic spatial fields in GSNs.

- Because qualitative properties form a smaller discrete domain than quantitative properties of space, processing and communication of qualitative information in GSNs can potentially be achieved more efficiently, using less resources, than for quantitative information.
- Any quantitative information generated by sensors nodes can always be converted into a less detailed qualitative representation, although the converse is not true. Further, the inherent imprecision of qualitative information can help make sensor networks more robust to imprecision and other forms of uncertainty in sensor readings.
- Using qualitative representations enables salient entities to be derived from complex dynamic fields, reducing system complexity and resulting in GSNs that are easier to design, construct, and query.

Looking at the problem from the application perspective, it is possible to identify at least five distinct issues facing any GSN for monitoring dynamic spatial fields in the environment.

1. Spatial: The phenomena of interest are spatial, for example involving points and regions; metric and topological relations; and spatial autocorrelation.
2. Temporal: The phenomena of interest change over time. Much of this change is spatiotemporal, including move-

ment and events, such as splitting and merging of regions within the field.

3. Scale dependency: The phenomena of interest are typically scale-dependent, and phenomena observable at one spatial scale may be different from those observable at a different scale.
4. Imperfect knowledge: Our knowledge of geographic information is always in some way imperfect. In particular, the spatial distribution of sensors and temporal frequency of sensing leads to *granularity*, the existence of grains or clumps in data.
5. Local computing: GSN are a type of highly distributed spatial computing, where individual nodes in the network typically only have access to information about their immediate vicinity. The challenge is to construct systems with desirable global properties using only local knowledge and local behaviors.

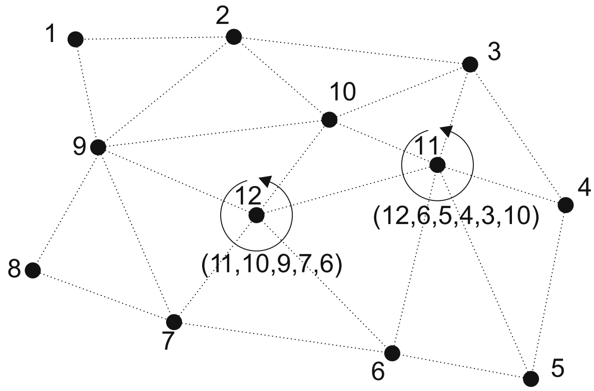
Although each of these five issues has a history of research in isolation, and in some combinations, the combination of all five of these issues makes monitoring of dynamic spatial fields especially complex. This complexity is fundamental to the need for qualitative representation and reasoning techniques. Representing continuous dynamic information using discrete sets of salient symbols is a first step to reducing complexity. The second step is to develop formal techniques for local reasoning about these discrete, salient symbols.

Example:
Event-Oriented Georesponsive Sensor Networks

As an example, imagine designing a sensor network tasked with monitoring a dynamic spatial field, such as a network for monitoring sea temperature in marine coral reef environments. Qualitative representation and reasoning can be used to managed the high levels of complexity in such application domains.

As explained above, the first step is to create discrete, qualitative representations of continuous information. Discretizing the values of the field itself starts to reduce complexity. Often, natural and salient categories may already exist for an application domain, such as a classification into "cold," "cool," "warm," and "hot" water. Using less detailed, qualitative representations can immediately increase robustness to imperfection into the system. For example, miscalibrated or inaccurate temperature sensors are much less likely to affect qualitative information about the temperature (e. g., "warm" versus "hot") than quantitative information (e. g., 35.2°C versus 34.9°C).

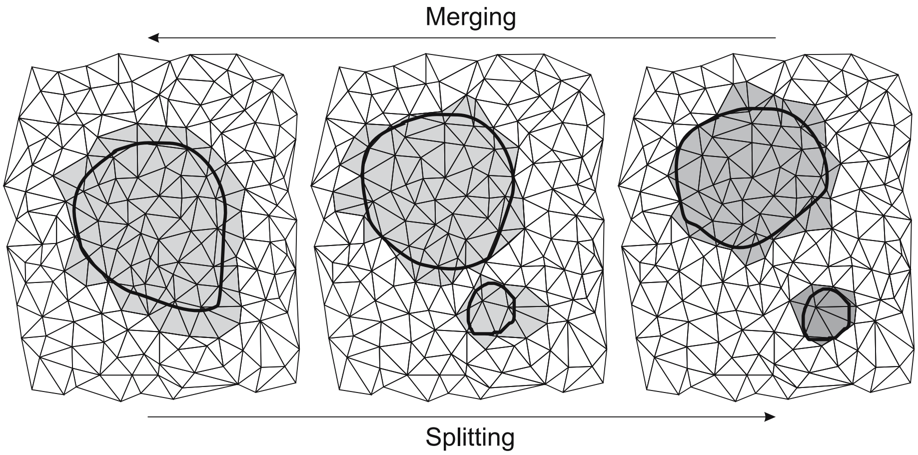
Discretization of the space itself further reduces complexity. The sensor nodes themselves already provide a discrete framework for the space under investigation. One approach to structuring this framework adopted by previous work [1,8] is to use a combinatorial map. Figure 1 illustrates the idea behind a combinatorial map, where each node has a number of defined neighbors and stores the cyclic ordering of those neighbors. Using a combinatorial map allows a variety of qualitative planar structures, such as triangulations, to be built without requiring any quantitative location information, such as geodetic or local coordinates. The structure is purely local: each node only knows information about the relative direction of it's immediate neighbors. Further, the inherent spatial imprecision in combinatorial maps, and related qualitative spatial structures, means that the resulting system can be more tolerant to imperfect information (e. g., the cyclic ordering of neighbors around a node is is much less likely to be subject to inaccuracy that, for example, location systems that rely on exact coordinate locations or bearings from one sensor to another).



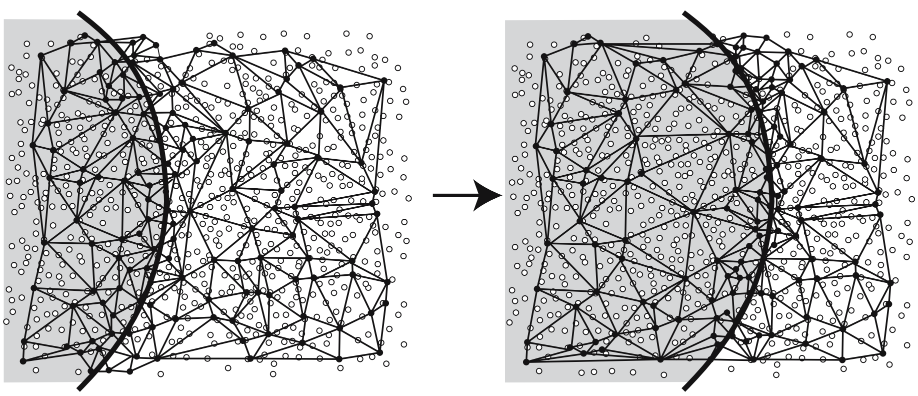
Geosensor Networks, Qualitative Monitoring of Dynamic Fields, Figure 1 Combinatorial map of sensor nodes and neighborhoods, including (counterclockwise) cyclic ordering for nodes 11 and 12

Having created appropriate qualitative representations, the second step is to develop local techniques for reasoning about these qualitative representations. In Fig. 2a region of dynamic spatial field (such as a high temperature "hotspot") is being tracked through a GSN structured as a triangulation (using a combinatorial map). Assuming the region moves continuously, a variety of qualitative spatial events can be locally detected. In order for regions to split or merge, for instance, they must first go through an intermediate stage where a single node connects two distinct parts (Fig. 2, center). As a consequence of the combinatorial map structure, this node can locally detect that a split/merge event is taking place (see [8] for more information).

Because the combinatorial maps is locally constructed, it can be efficiently and dynamically reconfigured. The concept of a *georesponsive* sensor network aims to acti-



Geosensor Networks, Qualitative Monitoring of Dynamic Fields, Figure 2 Local tracking of salient spatial events, such as splitting and merging



Geosensor Networks, Qualitative Monitoring of Dynamic Fields, Figure 3 Georesponsive sensor network, with increased node activation in vicinity of the boundary of large region

vate and deactivate sensors in response to changes in the dynamic field. Figure 3 illustrates the idea, where sensors in the vicinity of the boundary of the region of high temperature are activated to increase spatial resolution in those areas. Elsewhere, sensors are deactivated to increase sensor node lifetimes. Qualitative rules for achieving such behavior can be constructed based purely on qualitative spatial representations, like the combinatorial map (see [1] for further information).

Key Applications

Dynamic spatial fields are of interest across an enormous variety of environmental applications, including meteorology, land cover change, marine science, water resources management, defense, and emergency management and response. In general, applications of qualitative monitoring of dynamic spatial fields can fall into two broad categories. One category of use can be characterized as natural resource management, where decision makers use information gathered by GSN to manage scarce or fragile natural resources. Qualitative monitoring can help provide salient information to decision makers in a form that is more understandable and compatible with human conceptualization of dynamic spatial processes. Ultimately, such information can contribute to improved decision making. A second category of use can be characterized as scientific investigation of natural resources, where GSN are used by scientists to gather more detailed information about the environment than possible with conventional data logging techniques. In such cases, qualitative monitoring can assist in filtering data, screening out irrelevant data and highlighting high-level events of interest that can subsequently be investigated more closely.

Future Directions

As a relatively young area of study, qualitative monitoring of dynamic spatial fields has many important directions for future study, including:

- Sensor mobility: Although regions of a dynamic spatial field may be regarded as mobile, currently sensor nodes within the GSN are typically assumed to be static. Sensor mobility adds another layer of complexity to designing geosensor networks, which qualitative approaches are ideally suited to dealing with.
- Heterogeneity and multi-tasking: Current GSN usually comprise one type of sensor node engaged in a sin-

gle task. Future GSN will need to enable different types of node to interoperate on a variety of tasks, requiring the capability to integrate multiple qualitative queries across multiple different node types.

- Vagueness: Vagueness concerns the existence of boundary cases in information. Many qualitative concepts can be treated as vague and lacking precise boundaries (for example a "hot" region might be regarded as a vague spatial concept, if the boundary between "hot" and "not hot" is not precisely defined). Qualitative reasoning in the presence of vagueness remains an important challenge.
- Ubiquity: The goal of GSN is ultimately to embed spatial intelligence within the environment. Making GSN ubiquitous, unseen helpers that blend seamlessly requires the ability to manage complexity at every system level. Qualitative approaches provide one component of that complexity management, but further tools are required.

Cross References

► Distributed Geospatial Computing (DGC)
► Geosensor Networks, Estimating Continuous Phenonena
► Geosensor Networks
► Geosensor Networks, Formal Foundations
► Localization, Cooperative

Recommended Reading

1. Duckham, M., Nittel, S., Worboys, M.: Monitoring dynamic spatial fields using responsive geosensor networks. Shahabi, C., Boucelma, O. (eds.) ACM GIS 2005, pp. 51–60. ACM Press, New York (2005)
2. Egenhofer, M.J., Franzosa, R.D.: Point set topological relations. Int. J. Geogr. Inf. Syst. **5**, 161–174 (1991)
3. Frank, A.: Qualitative spatial reasoning about distances and directions in geographic space. J. Visual Lang. Comput. **3**, 343–371 (1992)
4. Freksa, C.: Using orientation information for qualitative spatial reasoning. In: Andrew U.F., Campari, I., Formentini, U. (eds.) Theories and Methods of Spatio-temporal Reasoning. Lecture Notes Computing Science vol. 639, pp. 162–178. Springer, Berlin (1992)
5. Galton, A.: Qualitative Spatial Change. Oxford University Press, Oxford (2000)
6. Hernández, D., Clementini, E., Felice, P.D.: Qualitative distances. In: Frank, A.U., Kuhn, W. (eds.) Proc. COSIT 95, Lecture Notes in Computer Science, vol. 988, pp. 45–57. Springer, Berlin (1995)
7. Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. In: Knowledge Rerpresentation, pp. 165–176. Morgan Kaufmann, San Francisco (1992)
8. Worboys, M.F., Duckham, M.: Monitoring qualitative spatiotemporal change for geosensor networks. Int. J. Geogr. Inf. Sci. **20**(10), 1087–1108 (2006)
9. Worboys, M.F.: Nearness relations in environmental space. Int. J. Geogr. Inf. Sci. **15**(7), 633–651 (2001)

Geospatial Analysis

► Data Analysis, Spatial

Geospatial Authorization

► Security Models, Geospatial

Geospatial Authorizations, Efficient Enforcement

VIJAY ATLURI, HEECHANG SHIN
Department of Management Science & Information Systems, Rutgers University, Newark, NJ, USA

Synonyms

Unified index scheme; Resolution-based matrix quadtree; Spatio-temporal-authorization-driven R-tree; Secure time-parameterized tree; Secure past, present and future tree

Definition

Enforcing security often incurs overheads, and as a result may degrade performance. The problem is exasperated more in geospatial data, which includes, among other things, geospatial image data and moving-object data. Uncontrolled dissemination of geospatial image data may have grave implications for national security and personal privacy. This is because high resolution satellite imagery may be used to identify vital national resources. As a result, this could encourage industrial espionage, terrorism, or cross-border military attacks, and combination of publicly available personal data pools with high resolution image data. This, coupled with the integration and analysis capabilities of modern geographic information systems, can result in a technological invasion of personal privacy. Similarly, the location and tracking of mobile users, required in effective delivery of location-based services, also raises a number of privacy and security issues, because disclosing the location information of mobile users has the potential to allow an adversary to physically locate a person for malicious purposes, and the location history can be used to identify the user's private information such as health status, political affiliations and religious beliefs. Thus, it is essential to have appropriate access control mechanisms in place. Unlike conventional authorizations that can be managed and searched using access control lists, management and searching of geospatial authorizations require a suitable indexing structure since they possess spatiotemporal attributes. As such, serving an access request requires the searching of two indexes: the index for data objects and the index for authorizations. However, the response time can be improved by using a unified index to support the user access requests. Here, the recently proposed unified index schemes are introduced. Specifically, two types of unified index are presented: (1) unified index structures for geospatial images, the resolution-based matrix quadtree (RMX-quadtree) and the spatiotemporal-authorization-driven R-tree (STAR-tree), and (2) unified index structures for moving objects, the secure time-parameterized R-tree (^{S}TPR-tree) and the secure past, present and future tree (S^{PPF}-tree).

Historical Background

Until recently, the access control model has not taken the spatial dimension into consideration. However, the need to control geographical information has increased because high-resolution satellite images have become readily available for public with reduced costs, and new context-aware applications such as location-based services have been developed due to the proliferation of mobile devices and wireless technologies. The *Geospatial Data Authorization Model* (GSAM) [1] is the first access-control model for geospatial data in the literature, and the model controls access to satellite images based on spatial extent, temporal duration, resolution, and other attributes of images. GSAM has been extended in two directions: one direction is support of vector-based spatial data [2,3], and the other is support of mobile data [4,5].

However, most of the proposed access-control models do not support the index structure of authorizations for efficient processing of geospatial access requests: for each access request, all the security policies that have been issued to the user of the access request are linearly searched and evaluated. Thus, in an environments where a high volume of user requests needs to be processed very fast, such as traffic management systems, the security enforcement process becomes the main bottleneck. Only a few authorization models that support the index scheme can be found in the context of a mobile database. An index scheme for moving-object data and user profiles has been proposed in [4]. However, this does not consider authorization. An index structure has been proposed to index authorizations ensuring that the customer profile information be disclosed to the merchants based on the choice of the customers [5]. However, this provides separate index structures for data and authorizations.

The RMX-quadtree [6] is the first proposal in the literature that allows geospatial images to be indexed based on their resolutions as well as their spatial attributes. Although the performance of the RMX-quadtree is efficient, it has two limitations: (1) the shape of the images to be indexed is restricted to a square, and (2) it does not allow overlapping images for a given resolution. The STAR-tree [7] eliminates these limitations. It is a three dimensional variant of R-tree, and it allows any overlapped rectangular shapes of images to be indexed.

In terms of moving-object data, the ^{S}TPR-tree [8] is an extension of the TPR-tree [9], and it is the first proposed unified index structure in literature for both moving objects and authorizations. One main limitation of the ^{S}TPR-tree is that it can only support those security policies based on the current and near-future locations of moving objects. Because the security policies in mobile environment are based on the *past*, *present* and *future* statuses of moving objects, ^{S}TPR-tree cannot fully handle security policies such as *track* because the past location of moving objects is not being stored. This limitation is resolved in the S^{PPF}-tree [10].

Scientific Fundamentals

Authorizations are typically implemented either as access-control lists or capabilities lists for traditional types of data. However, authorization specifications on geospatial data include a spatial dimension (such as authorized geospatial extents) and a temporal dimension (such as valid duration of time for access). Thus, in order to efficiently identify the authorizations that are relevant to an access request, an index structure can be used. Because geospatial data is also organized using index structures, a unified index structure that holds both geospatial data and authorizations can be created for efficient processing of user requests. The basic idea encompassing these unified index structures is to devise an appropriate index structure for the geospatial data, and then overlay authorizations on the relevant nodes of the tree. The construction of the tree and the overlaying process are performed in such a way that the cost of an access-control request is minimal. In the following, two types of unified indexing structures are presented: one for geospatial images, and another for moving objects.

Unified Index Structures for Geospatial Images

RMX-quadtree RMX-quadtree is a variant of the matrix (MX) quadtree: the structure of the RMX-quadtree permits overlaying of geospatial authorizations over nodes of the MX-quadtree. Thus, access requests for geospatial images can be processed more efficiently because only one index is used for evaluation. In order to build RMX-quadtrees, the following assumptions are made: (1) the spatial region represented by each image is a square, (2) images with the same resolution level are non-overlapping, and (3) higher resolution images cover smaller spatial extents.

The search space in the RMX-quadtree is assumed to be square and it is recursively decomposed into quadrants; the quadrants are named northwest (NW), northeast (NE), southwest (SW), and southeast (SE). An authorization is stored into a node *N* if the authorization applies to all the possible images in *N*'s subtrees. The structure satisfies the following properties:

- Images of the same resolution are stored at the same level.
- Each index node includes fixed number of children nodes (NW, NE, SW, and SE).
- Images can be stored in an index node as a result of merging different levels of images. Only the images with the highest resolution are stored in the leaf nodes.
- Each geospatial image corresponds to a 1×1 square, and it can be the spatial extent of index nodes.
- The depth of the quadtree is predefined and the structure is independent of the order of insertion operations.

Authorization specifications on geospatial data include spatial attributes and privilege modes containing geospatial operations. Therefore, to efficiently identify the authorizations relevant to an access request, an index structure is necessary. However, the index structure built based on authorizations would be unbalanced, because given two regions, if one region allows access only at lower resolution and the other region allows access at higher resolution, then the resultant index would be skewed due to the larger number of authorizations in the higher resolutions. Instead of trying to construct a tree for authorizations, RMX-quadtree chooses a different strategy, which is to overlay authorizations on top of the geospatial index tree, MX-quadtree. The overlaying means that the relevant authorization is stored on a node; thus, each node in the tree includes one additional field that points to the group of overlaid authorizations. This overlaying strategy guarantees the RMX-quadtree as a balanced tree because the base tree, the MX-quadtree, is balanced, and authorizations are simply stored on the relevant nodes of the tree; some of nodes may include more than one authorization, but this does not change the balance nature of the tree.

RMX-quadtrees are constructed by (1) creating MX-quadtrees for each resolution level of geospatial images, (2) merging them into one tree, and (3) overlaying authorizations on the tree. The first step is to build MX-quadtrees for each resolution level. Because MX-quadtrees are designed for storing only point data, an image is represented with its center location and edge length. Then, the MX-quadtree for the highest resolution level becomes the base tree, and other MX-quadtrees are merged with the base tree by adding the image data into the index node of the base tree. This is possible because the structure of the MX-quadtree is independent of the order of insertion operations, which becomes the property of the RMX-quadtree. Thus, the different levels of index nodes in the base tree refer to the same spatial region as the leaf nodes of other RMX-quadtrees. The last step is overlaying authorizations on the merged tree. An authorization is overlaid on a node when it applies to all the possible images in its subtrees. The reasoning for this overlaying process is that if a subject is allowed to access an area, it is also allowed to access all the regions at the same level of resolution within it. Representation of images using points creates a challenge for the overlaying process: an image stored in a node may not correspond to the area covered by the node. This is because the tree is constructed based on point information (center location of images) rather than geospatial regions that the image covers. In order to handle this situation, the area covered by a node *N* is expanded by $(p/2)$ where p is the edge length of *N*. The authorization is overlaid on *N* if, and only if, the spatial extent of the authorization fully encloses the expanded area of the node *N*. This overlaying process recursively traverses the tree and for each node, the above rule applies. The overlaying process halts when the traversal reaches the level of resolution of the authorization object or both of the spatial regions are disjoint. During the overlaying process, there are two possible cases. The first case is that an authorization is applicable to only parts of an image, and there exist no nodes in the tree that satisfy the overlaying rule; the second is that there is no spatial relationship between the authorization and the node, and thus, there is no reason to traverse down the tree further.

A user request for geospatial images includes the geospatial region of interest with the specific resolution level. The user request evaluation starts from the root node of RMX-quadtree. This process will fall into one of the following cases:

- Stopping evaluation of authorizations: this occurs when the node contains an authorization that includes the resolution level that the user requests, which means that the user is allowed access to all the images in the node's subtree, and therefore, no more authorizations need to be evaluated. However, the search process continues until it reaches the resolution level that the user requests, and returns the images that are enclosed in the spatial extent of the user request.
- Continuing to traverse the tree and check for more authorizations until reaching the node that stores same resolution level as the user requests.
- Halting the process; this happens when the geospatial region of the user request is disjoint with that of a node because all the images stored at the node are disjoint with the spatial region of the user request.

STAR-Tree A STAR-tree is a 3D variant of R-tree [11], and it is a unified index structure for geospatial images and authorizations similar to the RMX-quadtree. R-tree data structure is similar to B-tree, but different in the sense that it is extended to multidimensional space. The spatial extent of each node of R-tree is approximated by the minimum bounding rectangle (MBR) that encloses all its entries' spatial extents tightly, which implies that the spatial extent of the root node will enclose all the space that its children nodes embed. The node structure of R-tree can be summarized as follows: (1) a leaf node contains (a predefined number of) images and MBR, and (2) a non-leaf node contains (a predefined number of) child nodes (which can be another nonleaf nodes or leaf nodes) and MBR that tightly bounds all the MBRs of its child nodes.

By employing the capabilities of the R-tree, STAR-tree can index any rectangular shape of overlapping images, and therefore the native satellite images can be indexed without any preprocessing. In addition, it is capable of handling temporal attributes of the geospatial images: the capture time or the download time of images. Because STAR-tree can index spatiotemporal attributes of geospatial objects, security policies with a specific valid time interval for accessing the data can be supported.

To construct the STAR-tree, first the images of the same resolution are grouped together and a three-dimensional (3D) (x-, y-, and t-dimensions) R-tree is built for each group. Then, the tree with the highest resolution is taken as the base tree, and other trees are merged carefully in the order of their resolution, so that they do not violate the properties of the STAR-tree and until all the single-resolution trees are merged. Then authorizations are appropriately overlaid based on principles similar to those of the RMX-tree, essentially by comparing the bounding region of the node and the spatiotemporal region of authorizations. However, the process is more straightforward since STAR-tree is able to index 3D objects natively, and thus does not need to deal with the transformation between region and point as in the RMX-quadtree.

The following are the main properties of a STAR-tree:

- Unlike the RMX-quadtree, only leaf nodes store geospatial images.
- Images with the same resolution level are stored at the same level in STAR-tree.
- STAR-tree is an unbalanced tree, but this does not degrade performance because the tree operations (insert, delete, and search) will be processed only in the part of the tree from the root node to the leaf nodes with the specified resolution levels for tree operations, instead of the whole tree. This part of the tree is balanced with longer height compared to the R-tree for geospatial images for the specified resolution level.
- The best case for tree operations is $O(h)$, and $O(hm^{h-1}+N)$ is the worst case where h is the height of the STAR-tree, m the maximum number of entries that each node holds, M the number of authorizations overlaid on the tree, and N the number of images. Note that, since the RMX-tree is a more rigid tree structure that does not allow overlapping images, the cost is $O(k+M)$ where k is the height of the highest resolution and M the number of authorizations, which is not surprising.
- Similar to the RMX-tree, the images stored at a higher level in the tree will have lower resolutions. This is because images of lower resolution would cover more spatial region than those with higher resolution when their graphical size is the same.
- Compared to the R-tree, each node includes one additional field that points to the group of overlaid authorizations on the node.

To further improve the response time, authorizations can be preevaluated to compute the set of subjects involved in each authorization. Then the subjects associated with the authorizations overlaid on a node can be indexed using a B^+-tree. As a result, each node would include the B^+-tree whose key is a subject, and whose data is authorizations associated with the subject. This structure makes the evaluation process efficient because only the authorizations that are relevant to the user request would be evaluated instead of going through all the authorizations overlaid on the node.

Unified Index Structures for Moving Objects

^{S}TPR-Tree The ^{S}TPR-tree is an extension of the TPR-tree [9], and it is the first proposed unified index structure for both moving objects and authorizations in the literature. TPR-tree is a variant of R-tree and is designed to answer the queries for supporting present and future locations of moving objects. Because the locations of moving objects are constantly updated, the main challenge for a moving-object database is to minimize the updating cost. For this purpose, in the TPR-tree the moving object is represented as its initial location and its constant velocity vector; thus, a moving object is updated only if it deviates more than the specified tolerance level. This will reduce the necessity for frequent updating.

In order to support the moving-objects queries, a time-parameterized rectangle (TPR) is used for the same purpose of MBR in R-tree: time is parameterized in MBR so that for a given time, all the trajectories of moving objects stay in TPR. Also, if it is required to insert a new moving object, TPR-tree finds the minimum volume enlargement of TPR between the insertion time and the predefined duration of time, called the time horizon, because as time elapses, the future locations of moving objects become less accurate. Thus, the time horizon guarantees the validity of query results. The node structure of a TPR-tree is similar to R-tree: a leaf node contains (a predefined number of) locations of moving objects (represented as the combination of reference position vector and velocity vector), and a nonleaf node contains (a predefined number of) child nodes, which can be other nonleaf nodes or leaf nodes.

The security policies specify the access control rules to profiles and location as well as movement trajectory information of mobile users, or to stationary resources based on the mobile user's location. Thus, either a subject or an object in an authorization specification can be a moving object, which is being indexed.

The ^{S}TPR-tree is constructed via a consecutive insertion operation into an initially empty tree, then overlaying of authorizations over the nodes of the tree. It employs the same overlaying and user request evaluation processes as that of STAR-tree except the facts that (1) it does not need to handle resolutions (it can be considered a STAR-tree with only one resolution), and (2) TPR is used instead of MBR for the overlaying procedure. The spatiotemporal extent of TPR is bounded because the valid time duration of the node is finite, i. e., from the current time to the future time by the time horizon. Therefore, the TPR of a node is considered similar to the MBR of STAR-tree for overlaying and user request evaluation processes, and thus, same procedures as those of STAR-tree can be used.

One main limitation of the ^{S}TPR-tree is that it can only support security policies based on the current and future locations of moving objects. Because the security policies in mobile environments are based on the *past*, *present* and *future* statuses of moving objects, ^{S}TPR-tree cannot fully handle security policies such as *tracking* because the past status of moving objects is not being stored. This limitation is resolved in the S^{PPF}-tree using the concept of partial persistence. Another limitation is that the ^{S}TPR-tree is capable of overlaying authorizations where either subjects or objects in an authorization are moving objects, but not at the same time. This is because mobile subjects and objects would be stored in different nodes of the index, and thus, supporting such authorizations' overlaying may require splitting the subject and object components of the authorization.

S^{PPF}-Tree The previously introduced ^{S}TPR-tree cannot support security policies based on *tracking* of mobile users. It is important to note that tracking information could also be sensitive and therefore security policies are often specified to reflect this. To efficiently enforce these policies, the tree must support this functionality in the sense that all the location history of moving objects is preserved. S^{PPF}-tree, an extension of ^{S}TPR-tree, can maintain past, present and future positions of moving objects along with authorizations, by employing partial persistent storage. Thus, it can support security policies based on tracking of mobile objects.

S^{PPF}-tree is a variant of R^{PPF}-tree [12], which applies the concept of the partial persistence to the TPR-tree in order to preserve the past location of moving objects as well. Partial persistence is the key concept of R^{PPF}-tree, in order to keep past positions as well as present and future positions of moving objects. Observe that there are two kinds of moving objects: one is currently moving objects so that their final location is predicted but not decided (called *alive* moving objects), and another type is objects that have already stopped moving, or have changed their velocity or anticipated future location above the predefined deviation level (called *dead* moving objects). During update (insertion or deletion) of moving objects in the tree, the leaf node where the update occurs are evaluated to see if there still exists a prespecified range of alive moving objects. If the number is out of this specified range, alive objects in the node are copied into a new node (called *time split*). The original node is used for evaluating the past positions of moving objects; the newly created node is for the present and future positions of moving objects, as in ^{S}TPR-tree. A similar process is applied to index nodes: in this case, the number of alive children nodes is checked if it is within the predefined range.

Because S^{PPF}-tree maintains past positions of moving objects as well, the overlaying process is more complicated than that of the ^{S}TPR-tree because authorizations are required to be maintained properly not only for present and future positions but also past positions: in the case of ^{S}TPR-tree, the tree is reconstructed after some reasonable duration of time, and authorizations are batch-overlaid on the tree. Thus, there is no need to deal with maintenance of authorizations during the tree's lifetime. Since the S^{PPF}-tree handles all the history information as well, it is necessary to maintain the overlaid authorizations more carefully in order not to violate the overlaying strategy. An authorization log is introduced to handle this situation; whenever an authorization is applicable to the tree, the authorization log overlays the newly applicable authorization on the alive nodes, and relocates the authorizations from the alive nodes to the dead nodes if they are only applicable to the dead nodes. An authorization log is a data structure constructed by spreading all the authorizations on the time line. As time elapses, a new authorization becomes applicable to the tree when the valid time duration of the authorizations is overlapped with the tree's valid time duration, i. e., between the current time and the time horizon. Then, the authorization

G

log triggers an *auth_begin* event, which will overlay the authorization on the tree. On the other hand, certain overlaid authorizations become invalid when the valid time duration of the authorization is not applicable to the overlaid nodes. In this case, the authorization log triggers an *auth_end* event, which will remove the invalid authorizations from the overlaid nodes and reoverlay on the tree, because the removed ones may satisfy the overlaying conditions of other nodes in the tree. Also, an update must take care of the cases when the time-split occurs. Time-split creates a new node where some authorizations may be eligible to be overlaid on it. The authorization log supports a method, called *find-auth*, which computes all the authorizations overlapping with the valid interval of the newly created node. Then, the authorizations as a result of find-auth, will be overlaid on the new node if it meets the overlaying condition.

A user request evaluation is similar to that of ^{S}TPR-tree except that it can now evaluate a user request that also includes the tracking of moving objects, due to the functionality of holding all the updates history. In this case, only the nodes for which initial creation time and the time when time-split occurs, if time-splitted (otherwise, this time can be considered as current time) are overlapped with the time interval of the user request are evaluated.

Key Applications

Mobile Commerce Marketing

Owing to technological advances in mobile devices with wireless networks and positioning devices such as global positioning systems (GPS), customers' locations are used to provide customized services based on their current positions and movement histories. Mobile-targeted marketing or location-based advertising is one such example. Although there has been a general consensus that mobile-targeted marketing can provide more profits, most customers consider the advertised information spam unless they allow receiving advertisements. Therefore, the mobile service providers should employ a security scheme wherein users' preferences can be specified properly. Because customers are mobile, and their security policies are based on their current locations, tracking the locations of customers and enforcing security policies must be handled efficiently. Unified index schemes such as ^{S}TPR-tree and S^{PPF}-tree can handle the access request more efficiently than traditional approaches such as using two index structures (one for mobile customers and another for authorizations) because access-control requests are processed using just one index structure.

Selective Dissemination of High-Resolution Satellite Images

There are now more than 15 privately owned commercial satellites with resolutions from 1 to 30 m. For example, satellites such as IKONOS (launched in September 1999), ORBVIEW, EROS and QUICKBIRD are privately owned and provide images with resolution of 1 m or smaller. Uncontrolled dissemination of this information may have grave implications for national security and personal privacy, as some groups may exploit this information for aggressive purposes. Thus, formal policies for controlling the release of imagery based on geographical boundaries and resolutions of images have been proposed. However, the number of images being disseminated is tremendous. For example, in case of TerraServer-USA, the average daily image tiles transferred was 3,684,093 during 1998 and 2000. There is a need for effective and efficient schemes for facilitating controlled dissemination of satellite imagery. The proposed unified index schemes, such as RMX-quadtree and STAR-tree, can improve the response time because access requests can be processed more efficiently as only one index structure is used.

Securing Sensitive Resources

Physical location of individuals can be used to secure sensitive resources more effectively. For example, in order to gain the access to the secure repository room, an individual possessing relevant authorizations should be physically present in front of the room while she submits an access request. In order to do so, the location of mobile individuals and relevant security policies must be efficiently processed in large organizations. Unified index schemes such as ^{S}TPR-tree and S^{PPF}-tree can efficiently enforce security policies on mobile objects.

Traffic Management

Information on some traffic may be very sensitive knowledge to disclose because they are carrying dangerous materials. In this case, only an authorized group of people can locate and track the traffic. As the number of traffic and security policies increases, user requests must be handled efficiently. Unified index schemes such as ^{S}TPR-tree and S^{PPF}-tree can efficiently enforce security policies on the traffic management.

Future Directions

None of the above proposed unified index trees support negative authorizations. Providing such support is not trivial since they give rise to conflicts among the authorizations. Moreover, it may require changes to the fundamental assumptions used in the construction and access request evaluation. The overlaying strategy assumes only positive authorizations. Thus, an authorization is overlaid at as high a level as possible in the tree because as long as there exists an authorization that allows the user to access the given region, there will not exist any conflicting negative authorization that will not allow the user to access some parts of the allowed region. Based on this assumption, authorization evaluation halts whenever a relevant authorization is located during the traversal from the root node towards the leaf level. However, if negative authorizations are supported, all the authorizations overlaid on the traversal path need to be evaluated due to the possibility of conflicts among the authorizations: although an authorization that allows a user to access a region is overlaid in an index node, it is possible that another negative authorization that prohibits the user to access a part of the region may exist in the leaf node. Therefore, in order to support negative authorizations, it is necessary to create another copy of the data index and overlay positive authorizations on one index tree and negative authorizations on the other. Then, during the user request process, the result set from the positive authorization index is filtered out by the result set of the second negative authorization index.

Moreover, the ^{S}TPR-tree and S^{PPF}-tree cannot handle overlaying authorizations whose subjects and objects are both moving. As a result, supporting such authorizations' overlaying may require splitting the subjects and objects components.

Cross References

- ► Indexing Schemes for Multi-dimensional Moving Objects
- ► Indexing the Positions of Continuously Moving Objects
- ► Information Services, Geography
- ► Location-Based Services: Practices and Products
- ► Mobile Object Indexing
- ► Privacy Threats in Location-Based Services
- ► Privacy and Security Challenges in GIS
- ► R*-tree
- ► Raster Data
- ► R-Trees – A Dynamic Index Structure for Spatial Searching
- ► Security Models, Geospatial

Recommended Reading

1. Atluri, V., Chun, S.: An authorization model for geospatial data. IEEE Trans. Dependable Sec. Comput. **1**, 238–254 (2004)
2. Belussi, A., Bertino, E., Catania, B., Damiani, M., Nucita, A.: An authorization model for geographical maps. In: GIS Washington, DC, USA, 12–13 Nov 2004
3. Bertino, E., Damiani, M., Momini, D.: An Access Control System for a Web Map Management Service. In: RIDE, Boston, MA, USA, 28–29 March 2004
4. Atluri, V., Adam, N., Youssef, M.: Towards a unified index scheme for mobile data and customer profiles in a location-based service environment. In: Proceedings of theWorkshop on Next Generation Geospatial Information (NG2I'03), Cambridge, MA, USA 19–21 Oct 2003
5. Youssef, M., Adam, N., Atluri, V.: Preserving mobile customer privacy: an access control system for moving objects and customer information. In: International Conference on Mobile Data Management, Ayia Napa, Cyprus, 9–13 May 2005. Lecture Notes in Computer Science (2005)
6. Atluri, V., Mazzoleni, P.: Uniform indexing for geospatial data and authorizations. In: DBSec, Kings College, Cambridge, UK, 28–31 July 2002
7. Atluri, V., Guo, Q.: STAR-Tree: An index structure for efficient evaluation of spatiotemporal authorizations. In: IFIP TC11/WG 11.3 Eighteenth Annual Conference on Data and Applications Security, Sitges, Catalonia, Spain, 25–28 2004
8. Atluri, V., Guo, Q.: Unified index for mobile object data and authorizations. In: ESORICS, Milan, Italy, 12–14 2005
9. Saltenis, S., Jensen, C., Leutenegger, S., Lopez, M.: Indexing the positions of continuously moving objects. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, Texas, USA, 14–19 2000
10. Atluri, V., Shin, H.: Efficient enforcement of security policies based on tracking of mobile users. In: DBSec, Sophia, Antipolis, France, 31 Aug 2006
11. Guttman, A.: R-trees a dynamic index structure for spatial searching. In: ACM SIGMOD Conference, Boston, Massachusetts, 18–21 1984
12. Pelanis, M., Saltenis, S., Jensen, C.S.: Indexing the past, present and anticipated future positions of moving objects. ACM Trans. Database Syst. **31**(1), 255–298 (2006). doi: 10.1145/1132863.1132870

Geospatial Computational Grid

► Grid, Geospatial

Geospatial Computing Grid

► Grid, Geospatial

Geospatial Data Alignment

► Conflation of Geospatial Data

Geospatial Data Grid

► Grid, Geospatial

Geospatial Data Reconciliation

► Conflation of Geospatial Data

Geospatial Metadata

► Metadata

Geospatial Ontology

► Geospatial Semantic Web: Applications
► Geospatial Semantic Web, Interoperability

Geospatial Semantic Integration

NAIJUN ZHOU
Department of Geography, University of Maryland
College Park, College Park, MD, USA

Synonyms

Interoperability; Semantic Web; Ontology; Similarity, semantic; Features, linguistic; Features, Physical; Similarity discovery; Similarity Representation; Category, geographic; RDF; OWL; Geospatial semantics

Definition

Semantics refers to the meaning of symbols such as words, graphs, and other representations that are employed to describe a real-world object. A geospatial object is defined with its spatial distribution and its attributes. Geospatial semantics indicate the meaning of geospatial terms as attributes of geospatial objects. Geospatial semantic integration provides a global view of diverse terms in different data sources. For example, a geospatial object is described as a *river* in one data source and a *water body* in another source; geospatial semantic integration reveals that the two terms fundamentally mean the same thing in the real world and merge them into a single term such as water body.

Historical Background

Semantic integration, as one task of data integration, has been investigated in the database community since database technology was introduced in the 1960s. Geospatial semantics is concerned with the geospatial representation of reality that is a fundamental question of geospatial information theory. Recently, geospatial semantic integration has received increasing attention because more and more geospatial applications involve diverse data sources employing different terms to describe the same or similar geospatial objects and because new technologies (e. g., Semantic Web, emergency management) demand efficient machine-based communication among data sources. A special meeting, *Interoperating GISs* [1], was held in 1997 with a section focused on geospatial semantics. Early research on semantic integration is found in [2,3,4].

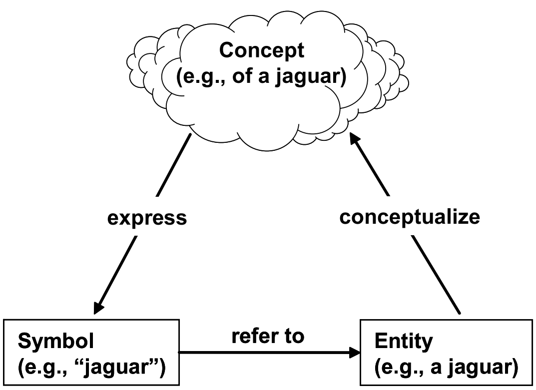
Scientific Fundamentals

Geospatial Semantics

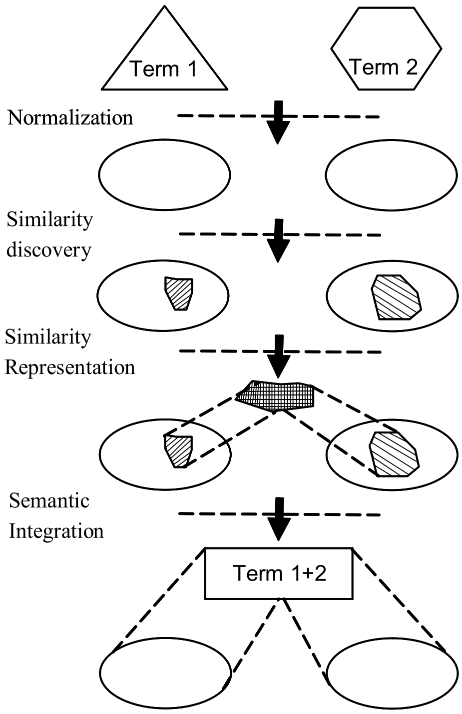
When humans describe and present an object, the object is first conceptualized and then is expressed in terms (Fig. 1). Therefore, the study of geospatial semantics involves the fields of spatial ontology and linguistics. Spatial ontology studies how human beings conceptualize an object, and linguistics investigates how the concepts are represented (e. g., as words).

Methods of Geospatial Semantic Integration

The primary tasks of geospatial semantic integration include understanding the semantics of geospatial terms and measuring their semantic relationships. Geospatial semantics can be obtained from the terms directly, or from additional information as the semantic definition of terms. It is well accepted that semantic integration should be based on semantic definition. Geospatial semantic integration can take a four-step approach: normalizing semantic definitions, measuring semantic similarity, representing and maintaining semantic similarity, and merging terms (Fig. 2) [6]. Semantic normalization may regulate the syntax of semantic definitions and may require that the definitions use linguistic words from the same thesaurus. Semantic similarity is a measure of semantic relationships, and the similarity can be quantitative (e. g., the probability that two terms indicate the same objects), or be qualitative values (e. g., superset, subset, equal, unequal, etc.).



Geospatial Semantic Integration, Figure 1 Semantic triangle [5]



Geospatial Semantic Integration, Figure 2 The four phases of semantic integration (adopted from [6]). Ellipses indicate semantics of terms

One geospatial semantic integration method deploys a feature-based approach originally proposed in psychology and cognitive science [7]. In addition to a geospatial term, a geospatial object also has a set of distinguishing features. Because the features define the meaning of the object, the comparison of the features provides semantic similarity of terms. The features can be physical or linguistic properties of geospatial objects. An example of physical features can be the *part*, *function* and *attributes* [8] of buildings; and two specific types of buildings (e. g., *stadium*, *theater*) can be compared and integrated based on their features. Linguistic features can be the words *branch*, *brook*, *canal*, *creek*, *ditch*, *gutter*, *river*, *rivulet*, *spillway* as the semantic definition of a *watercourse* (Spatial Data Transfer Standard, SDTS). The semantic similarity of the feature-based approach primarily measures how many common features two objects share [8,9,10].

Geospatial semantic integration can also be based on a common data model. Major common data models include Open Geospatial Consortium data standards, FGDC (Federal Geographic Data Committee) Content Standard for Digital Geospatial Metadata, SDTS, WordNet, etc. The common data model approach defines a common thesaurus containing the semantic definition of geospatial terms. Geospatial attributes can only take values from the predefined thesaurus from which semantic similarity is derived.

Ontology has been used in geospatial semantic integration [11]. Taxonomy is a light version ontology and is a major mechanism of defining geospatial terms. The use of taxonomy in geospatial semantic integration is supported by the theory of geographic category. Based on the theory of categorization, the perceived world is made up of structured information (i. e., taxonomy), and humans' categorization aims to reduce the differences among concepts to behaviorally and cognitively usable proportions. Geospatial semantic relationships are derived from taxonomy or from ontology mapping.

Geospatial semantics and ontology can be formally represented in XML, RDF, OWL, etc. Such representations enable the semantic communication among data sources to achieve automated semantic integration.

Key Applications

Geospatial Ontology Mapping

Geospatial ontology usually consists of a set of terms. Semantic integration provides a solution to the semantic relationships of the terms in different ontologies. In fact, semantic integration sometimes is also called ontology mapping (integration).

Geospatial Semantic Web

The communication of different ontologies and semantics are the primary tasks of the geospatial Semantic Web. Automated geospatial semantic integration enables machines to understand and exchange meanings and, thus, can help achieve the goals of the Semantic Web.

Geospatial Data Portals

Geospatial data portals allow users to search geospatial data sources that may use different terms for the same objects. Semantic integration provides a global view of diverse terms by comparing and merging terms. Such global view improves the efficiency and accuracy of data portal service.

Geospatial Database Integration

Geospatial database integration needs to merge schema and domain values [12,13]. The schema integration actually is also a task of semantic integration of database attributes.

Future Directions

Geospatial semantic integration is a much debated research area. There has not been a consensus on the definition of semantic integration. Current research and application activities mainly rely on the results from fields such as artificial intelligence, databases, cognitive science, linguistics, etc. Research is needed to address the uniqueness of geospatial information in defining and comparing the semantics of geospatial objects. There are two complimentary efforts to achieve geospatial semantic integration: defining common semantic models that regulate the use and exchange of semantics (top down approach), and developing automated methods to compare diverse semantics such that semantics can be defined independently by data providers. Finally, in addition to geospatial attributes, geospatial semantics should study geospatial expressions such as spatial topology and spatial operations; for example, the semantics of *close* should be interpreted as a certain spatial distance.

Cross References

► Geospatial Semantic Web

Recommended Reading

1. NCGIA (1997) Interoperating GISs. Report of a Specialist Meeting Held under the Auspices of the Varenius Project. December 5–6, 1997, Santa Barbara, California. http://ncgia.ucsb.edu/conf/interop97/interop_toc.html. Accessed 11 Sept 2007
2. Bishr, Y.: Overcoming the semantic and other barriers to GIS interoperability. Int. J. Geogr. Inf. Sci. **12**(4), 299–314 (1998)
3. Sheth, A.: Changing focus on interoperability in information systems: from system, syntax, structure to semantics. In: Goodchild, M., Egenhofer,M., Fegeas, R., Kottman, C. (eds.) Interoperating Geographic Information Systems, pp. 5–69 Kluwer Academic Publishers, Norwell, MA (1999)
4. Harvey, F., Kuhn, W., Pundt, H., Bishr, Y.: Semantic interoperability: A central issue for sharing geographic information. Ann. Reg. Sci. **33**, 213–232 (1999)
5. Warner, K.: Geospatial Semantics: Why, of What, and How?. J. Data Semant., Lect. Notes Comp. Sci. **3534**, 1–24 (2005)
6. Kalfoglou, Y., Hu, B., Reynolds, D., Shadbolt, N.: Semantic Integration Technologies Survey. CROSI Project, University of Southampton, Technical Report, E-Print No. 10842 May (2005)
7. Tversky, A.: Features of similarity. Psychol. Rev. **84**(4), 327–352 (1977)
8. Rodriguez, M., Egenhofer, M., Rugg, R.: Comparing geospatial entity classes: An asymmetric and context-dependent similarity measure. Int. J. Geogr. Inf. Sci. **18**(3), 229–256 (2004)
9. Kokla, M., Kavouras, M.: Fusion of top-level and geographical domain ontologies based on context formation and complementarity. Int. J. Geogr. Inf. Sci. **15**(7), 679–687 (2001)
10. Zhou, N.: A study on automatic ontology mapping of categorical information. In: Proceedings of National Conference on Digital Government Research, pp. 401–404. 18–21 May, Boston (2003)
11. Fonseca, F., Egenhofer, M., Davis, C., Camara, G.: Semantic granularity in ontology-driven geographic information systems. Ann. Math. Artif. Intell. **36**(1–2), 121–151 (2002)
12. Ziegler, P., Klaus, R., Dittrich, K.:Three Decades of Data Integration – All Problems Solved?". Paper presented at the 18th World Computer Congress, Toulouse, France, 22–27 Aug 2004. http://arvo.ifi.unizh.ch/dbtg/Staff/Ziegler/papers/ZieglerWCC2004.pdf. Accessed 11 Sept 2007
13. Doan, A.„ Halevy, A.Y.: Semantic Integration Research in the Database Community: A Brief Survey. AI Magazine **26**(1), 83–94 (2005)

Geospatial Semantic Interoperability

► Geospatial Semantic Web, Interoperability

Geospatial Semantic Web

FREDERICO FONSECA
College of Information Sciences and Technology,
The Pennsylvania State University,
University Park, PA, USA

Synonyms

Semantic web; Spatial semantic web; Ontologies

Definition

The Geospatial Semantic web addresses the geographic and spatial components of the Semantic Web, which is a project to create the infrastructure necessary for information sharing on the World Wide Web. The sharing should be based on the meaning of the documents. Since the interpretation of meaning is being done by computers another crucial issue is that whatever way this meaning is expressed it has to be machine-readable. Egenhofer highlights the need for a simple canonical form to pose geospatial data queries and methods to evaluate the semantics of the geospatial data sources to fit them to geospatial queries.

Historical Background

In his vision paper about the Semantic Web, Tim Berners-Lee (2001) gives the following example: "at the doctor's office, Lucy instructed her Semantic Web agent through her handheld Web browser. The agent promptly retrieved information about Mom's prescribed treatment from the doctor's agent, looked up several lists of providers, and checked for the ones in-plan for Mom's insurance within a 20-mile radius of her home and with a rating of excellent or very good on trusted rating services."

Both space and time can be found in this initial example. These are questions that are familiar to many geographic information science researchers. The future of the Semantic Web includes not only the geographic component but also the spatial component. For instance, Egenhofer addresses the combination of geospatial data with other spatial operators such as *in* for instance. Another perspective for the web user is to look exclusively for geospatial data. This second type of user is looking for images, maps, spatial databases, and tables in general. In this case, the geospatial data was previously classified as such. A third and widely available geospatial information resource on the web is the geographic descriptions present in personal web pages. The geospatial semantic web should be able to combine all these aspects of geographic information on the Web in a meaningful way.

Scientific Fundamentals

The main challenge for researchers in the geospatial semantic web is how to represent in computers what is special about spatial. Concepts of space and time that are intuitive for humans need to be carefully formalized in order to be used by computers.

In order to achieve the objective of the semantic web of exchanging information across disciplines with different world-views, it is necessary to have an agreement on the meaning of words—the categories, relations and instances that form parts of a mental model that in turn represents an understanding of the world. Such an explicit formalization of mental models is called an ontology. The basic description of the real things in the world—the description of what would be so-called objective truth—is called Ontology (with an upper-case O). The result of making explicit the agreement within communities is what the Artificial Intelligence community calls ontology (with a lower-case o). Therefore, there is only one Ontology, but many ontologies.

A geographic ontology has to provide a description of geographical entities, which can be conceptualized in two different views of the world. The field view considers spatial data to be a set of continuous distributions. The object view conceives the world as occupied by discrete, identifiable entities.

Web Services

The W3C definition of Web services is computer application that enables interoperable machine-to-machine interaction over a network. Such an application needs an interface described in a machine-processable format so that other application can interact with the service through the exchange of messages. The OpenGIS consortium has a standard for web services called OGC Web Services (OWS). OWS will define standards and procedures to create interoperable services that range from the acquisition of geospatial data from sensors to the creation of common infrastructure for decision support. Currently it has six major initiatives: (1) Sensor Web Enablement (SWE), designed to integrate a variety of sensors, platforms and management infrastructure into a single enterprise. The main focus of this initiative is on integration of physical sensors and simulators within a realistic operating environment; (2) Geo Processing Workflow (GPW), will interconnect geo-processes following workflow requirements; (3) Information Communities and Semantics (ICS) deals with complex geographic information types. The types will be modeled in UML and GML. After being catalogued, they will be available to be used in geo-processing workflows; (4) Agile Geography is targeted at the integration of processes and the enabling of services that show the possibilities of interoperability and service-oriented architectures using OGC Web Services. This initiative will apply links, bookmarks and Web pages to digital cartography and geospatial information management. It will also enable the integration of user initiatives interested in collaborating on the maintenance of a shared geospatial data sets; (5) Compliance Testing (CITE) will check for proof of compliance of Web Coverage Service and Catalog Service for the Web with OGC specifications. CITE will also require reference implementations of the specifications. Reference implementations are developed using open source standards; (6) CAD / GIS / BIM (CGB) has as an objective bridging the information models and workflows of the various communities involved with the 3-D representation cities. Future uses include location-based services and services of emergency preparedness and response. For this to be feasible it will be necessary to create a services-based architecture integrate information and make it accessible in a secure way.

This shows again the importance of the integration between OpenGIS and the W3C. The geospatial semantic web will need to build a semantic layer over the standards that the OpenGIS is developing or have already deployed.

Geography Markup Language

The OpenGIS Consortium has also created a language based on XML to store and exchange geographic data. The Geography Markup Language, GML (see http://www.opengis.org/docs/02-023r4.pdf for additional details), is able to encode OpenGIS geographic features. According to OGC, GML is "an XML grammar written in XML Schema for the modelling, transport, and storage of geographic information."

GML will be able to formally describe geographic features, coordinate reference systems, geometry, topology, time, and units of measure. OGC's view is based on features, which are abstractions of geographic phenomena associated with positions on the surface of the Earth. A computer representation of the Earth is a set of features.

Each feature can be described by a set of properties. Features may represent both vector data and coverages thus enabling a broader spectrum of information integration.

Queries

There are two important aspects to the queries created by users of the geospatial semantic web: the form and the presentation of the results. Egenhofer suggests a canonical form to pose geospatial queries:

<geospatial request> ::= <geospatial constraint>
[<logical connective> <geospatial request>]
<geospatial constraint> ::= <geospatial term>
<geospatial comparator><geospatial term>
<geospatial comparator>
::= ! based on the geospatial-relation ontology used
<geospatial term>
::= <geospatial class> | <geospatial label>
<geospatial class>
::= ! based on a geospatial feature ontology
<geospatial label> ::= ! based on a geospatial gazetteer

The second aspect in a geospatial query is how to represent the results. Depending on the nature of the query the result may be only an address such as an URI or real spatial data. What kind of spatial data and how to represent it is a question that an ontology of spatial data can help to answer. Again ontologies come into the picture. The solution for most semantic web problems involves the user committing to an ontology. This means that a user relies on some previous compilation of some explanation of facts that he/she thinks is truthful. In the query problem, the user will accept as a representation of the answer the representation of geographic data as stated in a specific ontology.

Geospatial Semantic Web Interoperability Experiment (GSWIE)

As a way of gathering all its initiatives, the Open Geospatial Consortium is developing the Geospatial Semantic Web Interoperability Experiment (GSWIE). According to OGC, the GSWIE "will address several important steps towards the development of a Geospatial Semantic Web (GSW), where discovery, query, and consumption of geospatial content are based on formal semantic specification. Many pieces of the GSW puzzle have been worked extensively in the last several years; this experiment aims to augment WFS/FE with a semantic query capability, through the definition of an ontology for the geospatial intelligence community." The experiment will address the creation of formal geospatial ontologies. It will also create and test service interfaces that can reference those ontologies and accept query requests in a semantic query language. It will be necessary to test also tools that will support the generation of semantically expressed geospatial information. The main goal of the experiment is to show that users may be able in the future to pose semantic queries to spatial databases of diverse sorts and receive meaningful answers in response.

Key Applications

The Geospatial Semantic Web is intended to handle information that is meaningful both for human and computers. Therefore, the way potential users understand and use information is very important. In particular, there are three basic dimensions for users of geographic information in the Geospatial Semantic Web context:

1. Professional: highly structured geographic information stored in geographic databases which are indexed, stored, or described in the Web;
2. Naïve: the retrieval of unstructured, subjacent, informal geoinformation in the Web;
3. Scientific: geographic information science papers, models, and theories available on the Web.

Professional: In order to improve the results of queries looking for information stored in geographic databases it is necessary to support better definition for spatial concepts and terms used across different disciplines and the development multiple spatial and terminological ontologies. Here there are also web services: the most common examples are the called location services: through the use of spatial databases, those services are able to locate restaurants, hotels, and other facilities depending on the user's location.

Naïve: This concept is based on Naïve Geography. This case looks for geographic information in the text of web pages. It can be pages with description of geographic features such as cities, geographic phenomena, or facilities. The user may be looking for complementary information or it can even be the main source of data.

A case in which it is possible to mix the content of textual web pages with more structured data such as the geographic distribution of machines on the Web is the ability to answer queries such as "I found this interesting web page, where is its geographic location?" or "Find other web sites that contain information about places close to places mentioned in this web site" or "List (or even graphically display) all the location information on the IBM web site (offices, research centers, etc.)". The user may also want to find pages that describe a special region of San Francisco and uses as a constraint that the web server in which the page is stored is located in San Francisco because he or she thinks that locals will give better naïve information.

Scientific: Here the situation is similar to what Google Scholar does today: a specialized search engine for scientific papers in a specific domain. Besides being able to index GIScience documents it is also necessary to create ontologies that express the different theories expressed in the papers.

Future Directions

The increasing generation and availability of geographic information brings with itself the possibility of using the Web as an easy and immediate channel for its distribution. For the semantic web to work it is necessary to efficiently index, retrieve, and integrate information on the web is the Semantic Web and its geographic counterpart, the Geospatial Semantic Web.

But it is important to recognize that the semantic web requires a level of organization and formalization that is not here yet. Furthermore, there is currently a wealth of geographic information on the Web that is available in different forms ranging from images, maps, spatial databases, and tables to locations of restaurants, description of landscapes, and informal reports of bird watching activities.

Research on the sharing and integration of geographic information through the Web vary from data distribution policy and legal issues, interfaces, performance, information retrieval, mapping, to information brokering and ontologies. Research on the Semantic Web is focused on building highly formalized representations of the available information resources called ontologies. A variety of tools and languages are currently available. But to capture the richness of the available geographic information on the Web it is necessary more than that. It is necessary to create representation of information resources that are more loosely organized. Another important step is how to link and map these more informal resources to the highly formalized representation structures being built by the Semantic Web. The result will be a broader spectrum of data available for users of geographic information. The specific context for this research relates to the understanding of the different kinds of geographic information on the Web and the corresponding representations (geo-ontologies) of these resources.

Recommended Reading

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web: A New Form of Web Content That Is Meaningful To Computers Will Unleash A Revolution Of New Possibilities. The Scientific American. **284**(5), 34–43 (2001)
2. Egenhofer, M.J.: Toward the Semantic Geospatial Web. Tthe Tenth ACM International Symposium on Advances in Geographic Information Systems, McLean, Virginia, pp. 1–4 (2002)
3. Fonseca, F.: From Geo-Pragmatics to Derivation Ontologies: New Directions for the GeoSpatial Semantic Web. Transactions in GIS. **11**(3), 313–316 (2007)
4. Fonseca, F., Egenhofer, M., Agouris, P., Câmara, G.: Using Ontologies for Integrated Geographic Information Systems. Transactions in GIS. **6**(3), 231–257 (2002)
5. Goodchild, M.: Geographical Data Modeling. Computers and Geosciences. **18**(4), 401–408 (1992)
6. Smith, B., Mark, D.: Ontology and Geographic Kinds. International Symposium on Spatial Data Handling, Vancouver, BC, Canada, pp. 308–320 (1998)
7. The Open Geospatial Consortium White Papers at www.opengeospatial.org/pressroom/papers

Geospatial Semantic Web: Applications

YASER BISHR
Image Matters LLC, Leesburg, VA, USA

G

Synonyms

Geospatial Web 3.0; Geospatial semantics web; Geospatial ontology

Definition

The semantic web is an extension of the current web. It is a vision for the future of the web, in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the web [1]. The geospatial semantic web adds space and time dimensions to the semantic web. "Web content" is used here to refer to network-accessible distributed digital contents. A geospatial semantic web application is any software that takes advantage of the geospatial semantic web. Ontology is a fundamental building block of the semantic web in general and the geospatial semantic web in particular. Ontology is a partial specification of conceptualization [2,3]. Ontology can be thought of as a data model about a domain of interest. The World Wide Web Consortium (W3C) developed and adopted the Web Ontology Language (OWL) as the standard language to encode and share ontologies.

Enriching geospatial contents with semantics is a two-step process. The first step is to create a geospatial ontology that captures the semantics of the domain. Geospatial ontology allows for definition and reasoning about real world spatial objects by combining information from multiple sources. A geospatial semantic approach to building applications allows for machines to understand and exchange qualitative spatial concepts like "eastbound vehicle" "near the airport" or "above ground" and hence complements the quantitative analytical power of geographic information systems (GIS). It also allows the definition and inference of higher level concepts like "highway" from lower level concepts like "paved road with speed limit 65 miles per hour". These concepts can then be instantiated to refer to actual objects in the database. The second step is to tag database elements with the ontology concepts so that they are accessible through the ontology and consumed by runtime geospatial semantic agents. These agents provide the runtime reasoning, querying, transformation and integration services upon which one can build geospatial semantic applications.

Historical Background

Historically, the use of GIS required specialized skills and therefore was limited to a small group of experts. Since then, GIS vendors and users have focused on adding analytical features to their GIS applications and have paid less attention to issues like interoperability, information sharing and integration. This situation has been further compounded over the last 30 years by an exponential growth of geospatial content, which has led public and private organizations to demand more interoperability, integration, and the ability to serve geospatial knowledge, rather than mere data. Perhaps the first published work on geospatial semantics is [4],which developed the ability to query heterogeneous geospatial database objects that are semantically tagged through a consistent and formal ontology. Some of the potential benefits geospatial semantic applications can provide are greater consistency and accuracy from improved system-wide management, more efficient use and sharing of knowledge, reduced redundancy of data across the system, better use of departmental GIS resources and reduced maintenance and support costs.

In recent years, the proliferation of the web and the latest advancements in what are broadly known as web mashups, which combine contents from different sources, have paved the way for general web users who are not GIS experts to capture, store, process, and display information about space and time. Egenhofer [5] naïve geography approach was proposed to mark the beginning of a research community push to facilitate the use of spatial concepts in nontraditional GIS applications and by general users.

Scientific Fundamentals

Geospatial semantic web applications allow users to use ontologies as a means to map (or tag) underlying data elements and their hidden relationships as explicit ontology concepts and predicates. In this way, users are able to unify data format and semantics regardless of their source. Through a high-level knowledge management interface that hides the complexity of geospatial content and operations, users can access and manipulate heterogeneous data in a unified semantic representation. It is important to note that geospatial semantic web applications are relatively new, and only a limited number of commercial off the shelf products exist. Academic research on the other hand is highly active in this field. Generally speaking, geospatial semantic web applications can be discussed from two distinct viewpoints. In the first, the focus is on the general architectural patterns, while in the second, the focus is on the geospatial ontologies that are consumed by those applications.

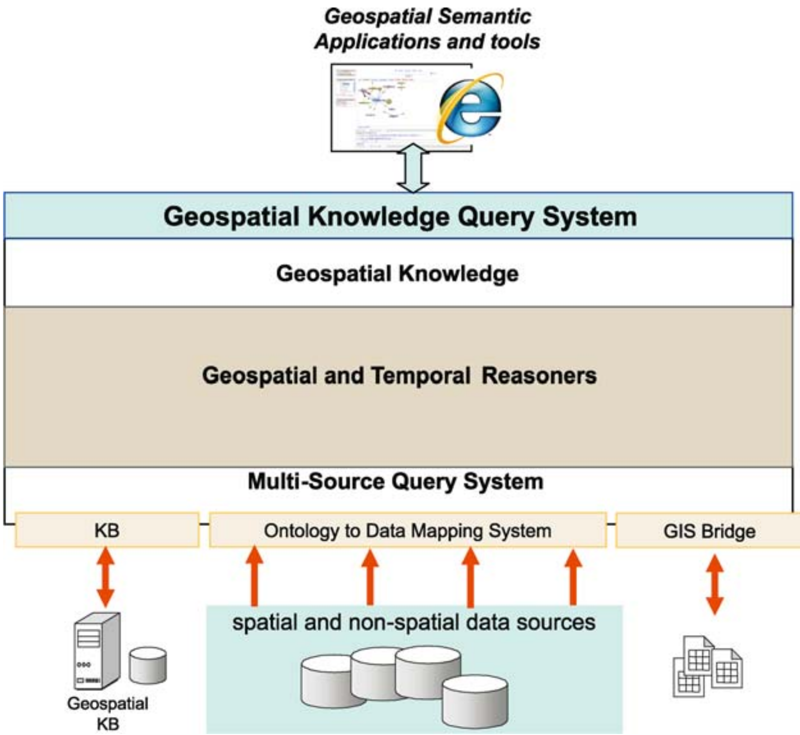
Geospatial Semantic Web Architecture View

A key benefit from geospatial semantic web technology is its ability to unify and fuse data semantics without resorting to changing the underlying data source schemas. This is contrasted to the prevailing data warehousing approach where relevant geospatial contents are mapped, transformed and migrated to a single large and data store. As depicted in Fig. 1, geospatial knowledge applications contain several components:

- Multisource inputs: a collection of semantically and syntactically heterogeneous and distributed spatial and non-spatial data sources.
- Ontology to data mapping system: a library of custom adapters providing consistent access to the underlying multi-source inputs. These adapters enable to explicitly map database features and their implicit relationships in and between the databases as ontological concepts and predicates respectively.
- GIS operators bridge: this is a component that facilitates the execution of traditional geospatial analytics. It returns the analysis results back the geospatial semantic system for further reasoning and querying.
- Multisource query engine: facilitates query processing and optimization; it translates queries based on the geospatial semantics to the underlying data sources.
- Spatial/temporal reasoners: a collection of spatial and temporal qualitative logical reasoners.
- Geospatial knowledge: the integrated virtual knowledge layer created from querying and reasoning about underlying data sources. This knowledge layer is specific to applications domains.
- Geospatial semantic query: provides semantic query interface so that applications can query to retrieve knowledge. This can trigger the reasoners and/or the multi-level query system. The W3C semantic WG has developed SPARQL for this purpose.
- Knowledge base (KB): a persistent knowledge base store that usually stores ontology information.

Geospatial Ontology View

A common underpinning for the semantic web as it is envisioned is that it contain several languages as shown in



Geospatial Semantic Web: Applications, Figure 1

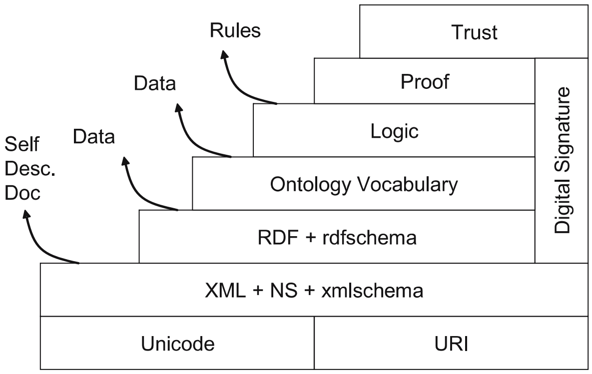
Fig. 2. The diagram depicts a semantic web architecture in which languages of increasing power are layered one on top of the other. The basis of a particular way of providing meaning for data is embodied in the model theory for the Resource Description Framework (RDF), OWL, and Logic Rules.

There are several proposals for describing space, spatial relationships, and relations between entities and their locations that have been developed within broad "ontological" frameworks. Several researchers have investigated the expressiveness of a number of publicly available ontologies. Similar to database schema design, those ontologies do not go beyond simple classifications and property associations. It is important to note here that developing ontologies must be executed with full knowledge of the kinds of inferences that geospatial semantic applications may require. To illustrate the above premise, an example of developing a model of nuclear facilities using XML schema (XML-S)and OWL is shown. Also shown are two different ways of representation in OWL which give completely different inference results.

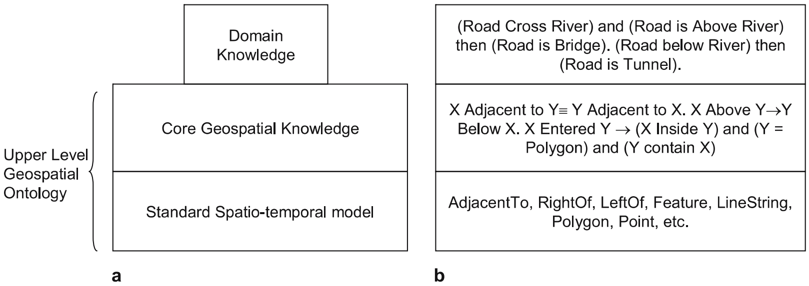
Example 1

```
<element name="ElectricSubstation"
type="ElectricSubstationType"/>
<complexType name=" ElectricSubstation Type">
<complexContent>
<extension base="Facility">
<attribute name="Produces"
type="anyURI"
default="#Electricity "/>
</extension>
</complexContent>
</complexType>
```

XML-S provides a standardized syntactical way to expose structural information. XML-S allows the definition of a schema for XML documents and provides some level of machine-understandable semantics of data. However, XML-S does not attach meaning to structural information.



Geospatial Semantic Web: Applications, Figure 2 Layers of the semantic web [Tim Berners Lee XML Conference, 2000]



Geospatial Semantic Web: Applications, Figure 3 Layered ontology for geospatial semantic web

As shown in Example 1, ElectricSubstation is a subclass of Facility and has a property Uses which has a default value: Coal.

Example 2

```
<owl:Class
rdf:ID="ElectricSubstation">
<rdfs:subClassOf
rdf:resource="#Facility"/>
<owl:Restriction>
<owl:onProperty
rdf:resource="#Produces"/>
<owl:hasValue
rdf:resource="# Electricity"/>
</owl:Restriction>
</owl:Class>
```

Example 2 conveys the same semantics as defined by the XML-S and does not necessarily add any value to the already defined XML-S.

Example 3

```
<owl:Class
rdf:ID="ElectricSubstation">
<owl:intersectionOf
rdf:parseType="Collection">
<owl:Class
rdf:about="#Facility"/>
<owl:Restriction>
<owl:onProperty
rdf:resource="#Produces"/>
<owl:hasValue
rdf:resource="#Electricity"/>
</owl:Restriction>
</owl:intersectionOf>
</owl:Class>
```

Example 3 shows an OWL representation, which essentially has added semantics. The members of the class are completely specified by the set operation. The example states that ElectricSubstation is the intersection of the class Facility and the set of things that produce Electricity. This means that if something produces Electricity and is a Facility, then it is an instance of ElectricSubstation. Without such a definition it would not be possible to know that Electricity Substations are facilities and they produce Electricity, but not vice versa.

The geospatial semantic web has at its core the basic semantic web layers, and it requires well-defined extensions to the ontology vocabulary and the logic layers. Constructing the core geospatial ontology is the most critical aspect for the success of geospatial semantic web. Application domains are expected to build their specific ontologies-based on the upper level ontology.

Figure 3 depicts a three-layered ontology for the geospatial semantic web:

- In the first layer a model of basic topologic, geometric types are defined, and the notion of features and feature properties are defined. A simple example of this layer is shown in Fig. 3b. The example above defines the concepts of *Adjacent, Above* and *Feature.* These concepts can be encoded in OWL.
- The default spatial inference and rules between features are defined in the core geospatial knowledge. A simple example of this layer is shown in Fig. 3b. Here *X Adjacent to Y* $\equiv$ *Y Adjacent to X* can be encoded in OWL. *X Above Y* $\rightarrow$ *Y Below X and X Entered Y* $\rightarrow$ *(X Inside Y) and (Y = Polygon) and (Y contain X)* are conjunctive rules that can be encoded in SWRL.
- The third layer is the application layer. These are specialized ontologies for specific applications, e. g., roads, rail, environmental engineering, etc. A simple example of this layer is shown in shown in Fig. 3b. In this example, applications define the concepts of Road, River and Bridge. These concepts can be defined in OWL. Also, applications define *(Road Cross River) and (Road is Above River) then (Road is Bridge), (Road below River) then (Road is Tunnel)* as conjunctive rules that can be written in SWRL.

Qualitative Spatial and Temporal Reasoning

The most common approach in most geospatial systems (e. g., GIS) relies on computational models of space, which involves a simple rendition of Euclidean geometry to compute spatial relations. This approach ties the spatial relations of features to their geometric rendition. For example if a land parcel is represented as a point on a map, it is not possible to indicate that it shares an edge with a gas station. For this to occur, land parcels must be encoded in the GIS as aerial features. Mathematical theories of space, upon which most GIS are built, has points as the primary primitive of spatial entities. Lines consist of collection of points, and areas, or regions, are defined as sets of lines and points. Simmons [6] rightly mentioned that "No one has ever perceived a point, or ever will do so, whereas people have perceived individuals of finite extent."

For the geospatial ontology to be useful, it is necessary to have some capabilities that can reason with it. As mentioned earlier, spatial and temporal reasoners derive answers from a knowledge base. They provide a methodology for reasoning about the information in the knowledge base, and for formulating conclusions. The knowledge base will naturally be the ontologies as defined before and the actual data instances. The geospatial semantic web is the marriage between GIS and the semantic web. It therefore complements GIS quantitative computational power by providing spatial and temporal qualitative reasoning. The principal goal of qualitative spatial reasoning is to make common-sense spatial knowledge explicit so that, given appropriate reasoning techniques, a computer could predict, diagnose and explain the spatial behavior of real world concepts in a qualitative manner without recourse to quantitative methods. Geometry focuses on quantitative aspects of space. Topology, on the other hand, answers qualitative questions about spaces, for example adjacency and directionality.

Elements of Geospatial Ontologies

Context Is the statement: "All Birds Fly" always true? Penguins are birds but can't fly. This statement, however, might be entirely true if its context was "in Brazil". Here, context was added to the statement, the context being "in Brazil". Several domains have already elaborated their own working definition of context. In a human–machine interaction, a context is a set of information that could be used to define and interpret a situation in which agents interact. In the context-aware applications community, the context is composed of a set of information for characterizing the situation in which humans interact with applications and the immediate environment. Context is considered here as the set of assertions and conditions under which a set of axioms are true. The geospatial ontology framework should indicate that a given set of ontology axioms can only be determined within context

Ontologies are shared models of a domain that encode a view which is common to different communities. Context is a model that cast a local view of shared models, i. e., shared ontologies. Context can be considered as a filter that helps scope this subset of an ontology that is relevant to a given situation. Developing a theory of context for the geospatial domains must satisfy the following requirements:

- Context should allow a simpler formalization of axioms by defining the set of known conditions that are common to the stated axioms.
- Context should allow us to restrict the vocabulary and the facts that are used to solve a problem on a given occasion. This requirement will enable us to scope large ontologies to those subsets that are relevant to the problem at hand.
- The truth values of facts should be dealt with as dependent on a collection of assumptions which implicitly define context.
- There are no absolute, context-independent facts, namely each fact must be stated in the appropriate context.
- Reasoning across different contexts should be modeled. This will enable mapping between ontologies (in context), and hence semantic interoperability.
- A theory of geospatial context must consider the role of time, location, and other spatial and temporal aspects in determining the truth value of a given set of axioms.

Identity Criteria (IC) IC determine *sufficient* conditions in determining the identity of concepts defined in ontology. IC may affect the taxonomic organization of ontology.

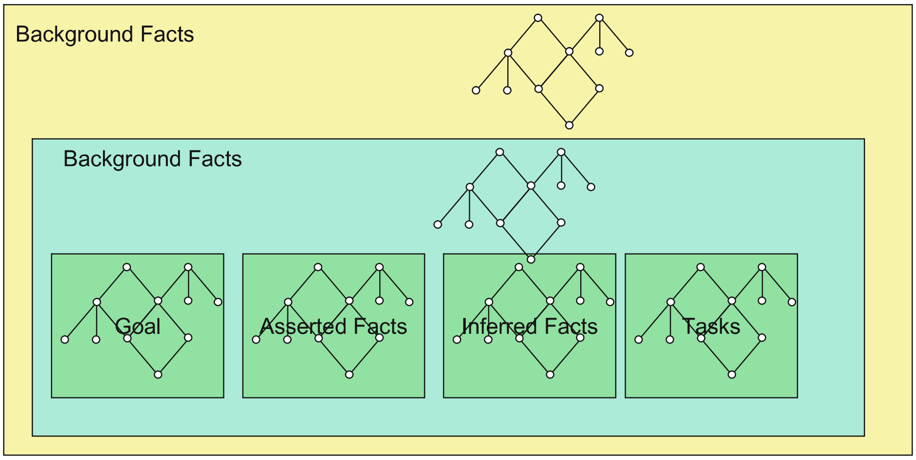
These are believed to be the most important criteria to create classification hierarchies in ontology. From an ontological perspective, IC is to:

- *Classify* an entity as an instance of a class C
- *Individuate* an entity as *distinct* instance of C
- *Identify* two entities at a given time *(synchronic identity)*
- *Re-identify* an instance of C across time (persistence, or *diachronic identity*)

One important issue to note here is the relationship between IC and location. Current GIS and even modern spatial ontologies adopt the premise that an object has to have some location, but that location is, in general, arbitrary. Just where an object can be is constrained by its physical constitution.

The above thesis flies against the realities of geospatial intelligence analysts. In many cases, analysts' determine the identity of objects by analyzing their relative location. For example, a tilted building in a high-resolution satellite image that covers Italy can only be the Leaning Tower of Pisa. On the other hand, a tilted building in Afghanistan can not be the Leaning Tower of Pisa.

G



Geospatial Semantic Web: Applications, Figure 4 Nested contexts

Following the above example, it then becomes important to always ask which spatial relationships can be used to define distinct ontological entities. Physical entities, for example, are typically distinguished from abstract entities precisely by virtue of their necessary location in time and space. In general, what is required is to be able to characterize precisely in what ways physical objects (or events) can be said to be located at particular locations in space. This raises questions about how the objects concerned are to be identified and how the locations are to be identified.

Spatial Reference System There are two distinct schools in representing location; absolutist and relativist position. Some researchers argue that for the purposes of linguistic semantics it is the relativist view that is compelling; another group favors the absolutist position, ignoring the fact that there is no such a thing as absolute reference system. This is because there is no single point from which the entire environment can be considered. Therefore, geospatial ontology GOF should be able to accommodate other views.

Mereotopology Mereology is the formal theory of part-relations: of the relations of part to whole and the relations of part to part within a whole. Mereotopology is the combined logical account of mereology and topology. Mereotopology is the theory of boundaries, contact, and separation built upon a mereological foundation. In the last few years, several researchers have shown that mereotopology is more than a trivial formal variant of point-set topology; Example 4 shows an example of mereotopological relationship.

Example 4

```
<owl:Class rdf:ID="RCCSpatialRegion">
<rdfs:label>RCCSpatialRegion
</rdfs:label>
</owl:Class>
<owl:ObjectProperty
rdf:ID="connectsWith">
<rdfs:label>connects
</rdfs:label>
<rdfs:domain
rdf:resource="#RCCSpatialRegion"/>
<rdfs:range
rdf:resource="#RCCSpatialRegion"/>
<rdf:type
rdf:resource="&owl;SymmetricProperty"/>
</owl:ObjectProperty>

<owl:ObjectProperty
rdf:ID="isDisconnectedFrom">
<rdfs:label>isDisconnectedFrom
</rdfs:label>
<rdfs:domain
rdf:resource="#RCCSpatialRegion"/>
<rdfs:range
rdf:resource="#RCCSpatialRegion"/>
</owl:ObjectProperty>.
```

The central aims of point-set and set-theoretic topology are to investigate and prove results concerning the properties of entire classes of spaces. On the other hand, mereotopology aims are to find perspicuous ways to represent and reason about the topological properties and relations of entities that exist in space.

Boundaries Geospatial semantic ontology should distinguishes between bona-fide and fiat boundaries. Bona fide boundaries are boundaries *in the things themselves*. Bona-fide boundaries exist independently of all human cognitive acts. They are a matter of qualitative differentiations or discontinuities of the underlying reality. Examples are surfaces of extended objects like cars, walls, the floor of a parking lot. Bona-fide boundaries are marked by bold solid lines in Fig. 2. Fiat boundaries exist only in virtue of different sorts of demarcation effected cognitively by human beings. Such boundaries may lie skew to boundaries of the bona fide sort as in the case of the boundaries of a parking spot in the center of a parking lot. They may also, however as in the case of a parking spot at the outer wall of the parking lot, involve a combination of fiat and bona-fide portions such as a wall at its back side. The classification of boundaries generalizes to a classification of objects. Bona-fide objects have a single topologically closed bona-fide boundary. Fiat objects have fiat boundary parts.

A prime example of the distinction between boundary types is colocation. Colocation of spatial objects means that they are located at exactly the same three- or two-dimensional region of space at the same moment in time. Colocation of boundary parts means that two- or one-dimensional boundary parts of spatial objects are located at exactly the same two- or one-dimensional region of space. Distinct bona-fide objects cannot be colocated since they cannot overlap. Owing to their atomic structure the surfaces of bona-fide objects can be brought in contact, i. e., the atoms forming their surface can come close but the atoms forming the surface of one object remain distinct from the atoms of the other. They do not mix and do not form a shared boundary. Distinct fiat objects of the same ontological kind cannot overlap. Bona-fide boundaries are dependent particulars. This approach stands opposed to the set-theoretic conception of boundaries as sets of independent points, each one of which could exist though all around it would be annihilated, and has a number of possible interpretations. The general statement of the fiat and boundaries is that one would be able to assert that the existence of any boundary is such as to imply the existence of some entity of higher dimension which it bounds.

Shape and Size In a purely qualitative ontology it is difficult to describe shape. In such an environment, very limited statements can be made about the shape of a region. Possible statements include whether the feature has holes, hollow, and whether it is one piece or not. One approach that has not been explored by the research community to qualitatively describe shapes is to use analogical reasoning and similarities. For example, one can say that the shape of this feature is analogous to a crown and then one can create assertions about crown-shaped objects.

Suppose that an intelligence analyst runs object extraction software on a satellite image. The software had recognized and identified an object as a sport utility vehicle (SUV). During the course of collecting other evidence about this object from other sources, the analysts get a more accurate dimension of the SUV size. Once the size of the SUV is entered, the system recognizes that these are not the dimension of a SUV, but of a truck used to carry troops and hence it reclassifies the object as a truck.

It is evident from the above example that size plays an important role in the identification of some features. Current GIS is supported by the belief that identity of features does not depend on some physical quantities. There are no GIS that can check the consistency of feature identification with respect to their size. However, the size of an entity can determine its identity: consider a lake versus a pond, an island versus a continent, or a car versus a SUV.

Key Applications

The geospatial semantic web is a horizontal capability that is still in its infancy. It has the potential to penetrate virtually all applications where reasoning about space and time are relevant. This includes but is not limited to:

- Location-based services and advertising
- Spatially enabled tags and folksonomy
- Enterprise information, application integration
- Semantic discovery and mash-up of heterogeneous geospatial content and geospatial web services
- Addition of location- and time-aware services to web-based social networks and other web-based consumer services
- Management of mobile and moving assets
- Workforce management

Future Directions

Geospatial semantic web research has been identified as a short-term priority in 2002. In the longer term the W3C had identified spatial ontologies as a long term research challenge. Key research areas include spatial and temporal reasoning, information retrieval, integration, and semantic ontology development. Geospatial information systems are becoming part of the mainstream IT. The geospatial semantic web is expected to be the main catalyst for such proliferation. Semantic web and GIS capabilities will be buried under the mainstream web technology stack. Semantic web inference engines will be able to natively support spatial and temporal reasoning support. Yahoo maps, Google Earth and many other similar web services have made geospatial web mash-up possible. The geospatial semantic web will surely facilitate automated service discovery and mash-up.

GIS applications are becoming ubiquitous. Geospatial semantic web technology will be the main facilitator that will enable developers to bury and hide GIS complexities under sophisticated and yet simple user interfaces, embedded in new consumer gadgets, or integrated with mainstream IT. Several research efforts are currently underway to standardize on core geospatial ontology to facilitate semantic interoperability.

Research directions from the scientific perspective are geospatial ontology core,the need for uncertainty and fuzzy reasoning, enterprise integration, semantic spatial services, and enterprise integration of spatial information.

Recommended Reading

1. OWL Overview http://www.w3.org/TR/owl-features/
2. Gruber, T.: A translation approach to portable ontology specifications. Knowl. Acquis. **5**, 199–220 (1993)
3. Gruber, T.: Toward principles for the design of ontologies used for knowledge sharing. Int. J. Hum.-Comput. Stud. **43**, 907–928 (1995)
4. Bishr, Y.: Semantic Aspects of GIS. ITC, (1997)
5. Egenhofer, M.: Naïve geography. In: Frank, A., Kuhn, W. (eds.) COSIT '95, Semmering, Austria, September 1995. Lecture Notes in Computer Science, vol. 988, pp. 1–15, Springer-Verlag, (1995)
6. Simmons, P.: Parts: A Study in Ontology. Clarendon, Oxford (1987)
7. Egenhofer, M.: Toward the semantic geospatial web. In: Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems,
8. Resource Description Framework http://www.w3.org/RDF/
9. Rule Markup Language (2002) http://www.ruleml.org/
10. Semantic Web Rule Markup Language http://www.daml.org/2003/11/swrl/
11. Web Ontology Language http://www.w3.org/2004/OWL/

Geospatial Semantic Web, Definition

BHAVANI THURAISINGHAM, LATIFUR KHAN, GANESH SUBBIAH, ASHRAFUL ALAM, MURAT KANTARCIOGLU
Department of Computer Science, The University of Texas at Dallas, Dallas, TX, USA

Definition

Geospatial semantic web refers to an intelligent, machine-understandable web where geospatial data are encoded in a semantic-rich data model to facilitate automated decision making and efficient data integration.

Main Text

Geospatial semantic web is a set of enabling technology that promotes more efficient use of geospatial data. The loose structure of the provided data model ensures a flexible environment for applications outside of a pure geospatial domain to freely mix geospatial data with native data without having to know all the details ahead of time. The geospatial data management gets a boost from the inherent data model and logic-based reasoning framework, since data can be filtered more efficiently upon acquisition. Moreover, implicit knowledge can be discovered through inference. The hundreds of geospatial clearinghouses that have stored volumes of data over the years but cannot disseminate them very effectively due to security vulnerability and data assurance issues can now utilize the semantic web concepts to solve their problems.

Geospatial Semantic Web, Interoperability

LIPING DI, PEISHENG ZHAO
Center for Spatial Information Science and Systems (CSISS), George Mason University, Fairfax, VA, USA

Synonyms

Semantic geospatial web; Geospatial semantic interoperability; Geospatial ontology

Definition

According to ISO/IEC 2382-01, *Information Technology Vocabulary, Fundamental Terms*, interoperability is defined as *"the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units"*. Based on various forms of perspective on information heterogeneity, interoperability can be classified in four different levels [1].

1. System interoperability involves bridging the differences in hardware, system software and communication systems.
2. Syntax interoperability aims to solve the differences in machine-readable aspects of data representation.
3. Structure interoperability constructs a uniform view of data modeling.
4. Semantic interoperability is concerned with having a common understanding of the meaning of the exchanged information in different systems.

The interoperability within the geospatial semantic web is the capability of sharing domain-specific resources and knowledge among the domains of geospatial science in the semantic web environment. There is no general agreement on what exactly constitutes semantic interoperability in the geospatial semantic web. Some elements of semantic interoperability include: 1) top of the interoperability stack for distributed computing; 2) shared concepts; 3) formal, common knowledge representation, usually in graph form, generically termed "ontologies"; 4) actionable (i. e., machine-process-able) relationships between similar/connected concepts using inference, reasoning, pattern matching, rule processing, etc.; and 5) shared models for distributed processes, such as interface contracts and virtual machines [20]. The interoperability in the geospatial semantic web facilitates the cross-domain resource discovery, heterogeneous resource query, and automatic resource translation from one domain to another.

Historical Background

With the rapid development of Web (e. g., Web Service) and distributed computing infrastructures (e. g., GRID), it is easy to access a large amount of independently developed and managed information resources of a broad variety. The success of syntactically interoperable Web service connections has created significant semantic gaps in what can be utilized. Both users and their client software may lack the tools to work with the diverse information which Web service standards have made nominally obtainable. *"The Semantic Web seeks to make the meaning as accessible as the material by enabling connections which are both logical and (machine) actionable between concepts which a user presently understands and those which may be new and foreign"* [21]. Interoperability in the semantic web is the single most important ingredient for satisfying the increasing need to reuse and analyze data in order to create information and knowledge and to subsequently reuse and share it across multiple domains because it makes the meaning understandable by broad user communities.

Tim Berners-Lee, the inventor of the Web, is the driving force of the Semantic Web initiative [2]. Recently a growing number of individuals and groups from academia and industry have been evolving the Web into another level – the Semantic Web, in which the meaning of information plays a far more important role. By representing not just words, but their definitions and contexts, the Semantic Web uses descriptive technologies (e. g., RDF and OWL) to provide a common semantic interoperable framework in which information is given a well-defined meaning such that distributed heterogenous the data and applications can be interoperable regardless of their differences on system, syntax, structure and semantics.

Because of the multi-source, multi-format, multi-scale, and multi-disciplinary nature of geospatial data and processing, the importance of semantics on accessing and integrating distributed heterogeneous geospatial information has long been recognized. The advances in the Semantic Web promise a generic framework to use ontologies to capture information meanings and relationships. Therefore, without doubt, adopting Semantic Web technologies to solve the issues of geospatial semantic interoperability is a good choice. Parallel to the development of the Semantic web, momentum has gained in recent years for the development of the Geospatial Semantic Web (GSW) – a geospatial domain-specific version of the Semantic Web which gives both people and machines true access to a wider range of geospatial knowledge. The GSW provides a common interoperable framework in which geospatial information is given a well-defined meaning such that distributed heterogeneous geospatial data and applications can be interoperable regardless of their differences on system, syntax, structure and semantics.

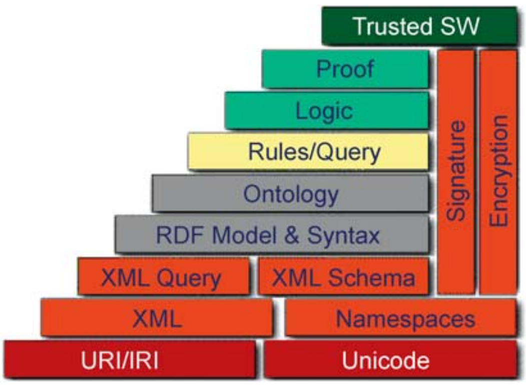
The Semantic Web provides a generic framework for semantic interoperability. It does not explicitly address some of the most basic geospatial entities, properties and relationships that are critical to geospatial information processing tasks, such as direction, orientation, spatial aggregation and topology. To better support the discovery, retrieval and consumption of geospatial content, the GSW creates and manages geospatial ontologies to exploit the logical structure of the geospatial world and allows applications to take advantage of "intelligent" geospatial reasoning capabilities. It does so by incorporating geospatial data semantics and exploiting the semantics of both the processing of geospatial relationships and the description of tightly coupled service content [3,4].

The Geospatial Semantic Web was listed as an immediately-considered research priority in 2002 by the University Consortium for Geospatial Information Science (UCGIS) [5]. Three eminent research questions were identified—creation and management of geo-ontologies, matching geographic concepts in Web pages to geo-ontologies, and ontology integration. Subsequently, a number of geospatial ontologies have been developed, such as the ISO19100 series, RDF geo vocabulary, OWL-Space, and NASA GCMD. In collaboration with industry, government and universities, the Open Geospatial Consortium (OGC) conducted the Geospatial Semantic Web Interoperability Experiment (GSW-IE) in 2005. GSW-IE aims to develop a method for discovering, querying and collecting geospatial content on the basis of formal semantic specifications. Significant achievements of this experiment include [21]: 1) development / encoding of formal geospatial ontologies, including feature type descriptions, 2) geospatial service interfaces which can provide service information formulated in the OWL-S semantic expression language referencing those ontologies, 3) OGC Web Feature Service interfaces which can operate on requests for ontologically expressed service and content descriptions, 4) semantic query language processing interfaces, and 5) tools for generating semantically expressed geospatial information.

Scientific Fundamentals

Compared to the conventional Web, the Semantic Web in advanced in two aspects: 1) common formats for data interchange (the original Web only has interchange of documents), and 2) a language for recording how the data maps to real world objects [6]. With such advances, intelligent systems with the help of reasoning engines and Web-crawling agents can go one step further to answer questions such as "*which airfields within 500 miles of Kandahar support C5A aircraft?*", rather than simply returning Web pages that contain the text "airfield" and "Kandahar", which most search engines can do today.

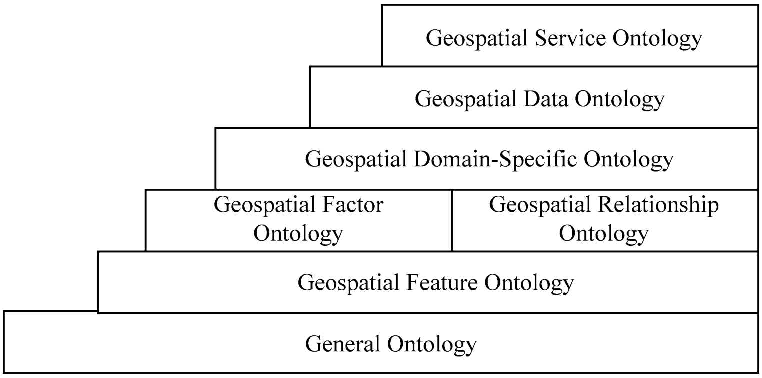
The hierarchical architecture of the Semantic Web is shown in Fig. 1. At the bottom level, Extensible Markup Language (XML) is a data-centric, customizable language providing syntax to represent structured documents with a user-defined vocabulary, however, it imposes no semantic constraints on the meaning of these documents. XML schema provides a means for restricting the structure of XML documents. Resource Description Framework (RDF) is a basic data model with XML syntax for referring to objects ("resources") and how they are related in order to allow information exchange between applications without loss of meaning. RDF Schema is a semantic extension of RDF for describing the properties of generalization-hierarchies and classes of RDF resources. RDF Schema can be used as a primitive ontology language. Web Ontology Language (OWL) adds more vocabularies for explicitly representing the meaning of terms and the relationships between those terms, such as relations between classes (e. g., disjointness), cardinality (e. g., "exactly one"), equality, richer typing of properties, the characteristics of properties (e. g., symmetry), and enumerated classes. The logic layer represents and derives knowledge, and the proof layer involves deductive process and proof validation. A digital signature can be used to sign and export the derived knowledge. A trust layer provides the trust level or its quality rating. Such a trust mechanism helps in building users' confidence in the process and the quality of the information provided and prevailing the Semantic Web [8]. The architecture of the GSW is similar to the SW portrayed in Fig. 1, but with extensive domain-specific technology and knowledge. The GSW shares the basic components with the Semantic Web, such as top level (general) ontology, ontological languages, and general reasoning mechanisms. However, it extends the Semantic Web with domain-specific components. Among all the components of the GSW, two are especially unique – geospatial ontology and geospatial reasoning. The former aims at expressing geospatial concepts and relationships, specialized processing of geospatial rules and relationships, and self-described geospatial Web service in a formal way. The latter embraces sets of geospatial inference rules on the basis of geospatial ontologies and techniques to conduct automated geospatial reasoning for deriving geospatial knowledge by machine with less human interaction.



Geospatial Semantic Web, Interoperability, Figure 1 W3C Semantic Stack [7]

Ontology is the centerpiece of the most prevalent semantic technologies. It enables the semantic interoperability by providing a formal way of representing, acquiring and utilizing knowledge. Ontology originated from philosophy as a reference to the nature and the organization of reality. In general, an ontology is a "*specification of a conceptualization*" [9]. In the domain of computer science, ontology provides a commonly agreed understanding of domain knowledge in a generic way for sharing across applications and groups [10]. In more practical terms, ontology consists of a list of vocabularies and the relationships between them. The vocabularies denote the domain concepts (*classes* of objects), such as aircraft or runway. The *relationships* indicate hierarchies of concepts. They also include property information (an airfield has runways), value restrictions (aircraft can only take off at an airfield), disjointness statements (aircraft and train are disjoint), and specification of logical relationships between objects (a runway must be at least 400 meters long for supporting C5A aircraft).

In comparison to general ontologies, geospatial ontology specifically describes 1) spatial factors, e. g., location and units; 2) spatial relationships, e. g., inside, near, and far; 3) physical facts, e. g., physical phenomena, physical properties, and physical substances; 4) disciplines, e. g., scientific domains and projects; 5) data collection, e. g., collection properties, such as instruments, platforms and sensors; and 6) geospatial computing models, e. g., input, output and preconditions. A well-formatted geospatial ontology is the core for enabling geospatial semantic interoperability.



Geospatial Semantic Web, Interoperability, Figure 2 The Hierarchy of Geospatial Ontology

It enables the following key functions in geospatial semantic Webs:

- **Communication** among humans and information systems. Since the geospatial sciences is about the research across a variety of scales and disciplines, the semantics of geospatial information is essential for the development of interoperable geospatial software and data formats. Geospatial ontology provides a common understanding of not only general geospatial concepts but also in context of geospatial scientific computing
- **Spatial reasoning** about geospatial associations and patterns, e. g., topological relations (connectivity, adjacency and intersection of geospatial objects), cardinal direction (relative directions among geospatial objects, such as east, west and southwest), and proximity relations, such as A is close to B and X is very far from Y), and contextual relations, such as an obstacle separates two objects that would be considered nearby space, but are considered far because of the obstacle [11].

Based on the interactions and the role within the context of the GSW, geospatial ontology can be classified into several large groups. The hierarchical relationships of those groups are shown in Fig. 2.

General ontology is the core upper level vocabulary representing the common human consensus reality that all other ontologies must reference. There are several well-developed general ontologies. The Dublin Core Metadata [12] is a standard of metadata vocabularies describing resources that enables the development of more intelligent information discovery systems. OpenCyc [13] defines more than 47,000 upper level concepts and 306,000 assertions about these concepts.

Geospatial feature ontology, defining geospatial entities and physical phenomena, forms the ontological foundation for geospatial information. It should be coordinated with the development of geospatial standards. The ISO 19100 series and the OGC specifications are very helpful in defining its scope and content.

Geospatial factor ontology describes geospatial location, unit conversions factors and numerical extensions. Geospatial relationship ontology represents geospatial and logical relationships between geospatial features to enable geospatial topological, proximity and contextual reasoning. The RDF geo vocabulary [14] provides a basic RDF vocabulary with a namespace for representing latitude, longitude and other information about spatially-located things using WGS84 as a reference datum. The OWL-Space initiative (formerly DAML-Space) provides the ontologies of comprehensive spatial properties and relations. By incorporating Region Connection Calculus (RCC), the CoBrA [15] and the SWETO-GS [11] define basic relations between two-dimensional areas and relevant rules to enable the reasoning of spatiotemporal thematic proximity.

Geospatial domain-specific ontology represents the domain concepts by using proprietary vocabularies. Sometimes there are some different representations for a geospatial feature in different domains. To achieve interoperability, there must be a link between domain ontology and feature ontology, either by subclassing feature ontology concepts or by mapping from feature concepts to domain concepts. To provide formal semantic descriptions of NASA data sets and scientific concepts, several projects are underway to develop a semantic framework. Defined in OWL, the ontologies within the Semantic Web for Earth and Environmental Terminology (SWEET) [16] contain several thousand terms spanning a broad extent of Earth system science and related concepts (such as NASA GCMD, ESML, ESMF, grid computing, and OGC). The SWEET provides a high-level semantic description of Earth system science.

Geospatial data ontology uses the ontologies below it, as seen in Fig. 2, to provide a dataset description including representation, storage, modeling, format, resources, services and distributions. It ensures that the geospatial data are discoverable, usable, and interoperable semantically.

The ontologies of geographic metadata, such as ISO 19115 ontology [17], add semantic meanings and relationships to the data description by which data sets are explicitly associated with providers, instruments, sensors and disciplines. Geospatial service ontology describes who provides the service, what the service does and what other properties the service has that make it discoverable. It also states the service inputs, outputs, preconditions and effects to ensure the usability of the service. The inputs and outputs of a service are data described by data ontology. To allow dynamic invocation, this ontology includes the concrete service ports, protocols and encodings.

Key Applications

With geospatial ontology-based semantic interoperability, the GSW promises an "intelligent" method to discover, retrieve, integrate, and utilize large and diverse geospatial information and services over the Web. Numerous efforts to approach this "intelligence" are currently active.

Intelligent Search Engine

SWEET is a prototype for showing how the Semantic Web can be implemented in Earth Science. By incorporating the Global Change Master Directory (GCMD), SWEET provides 11 ontologies to describe Earth and environmental concepts and their relations, including Earth realm, nonliving element, living element, physical property, units, numerical entity, temporal entity, spatial entity, phenomena, human activities and data. Aided by these ontologies, an intelligent search engine (http://sweet.jpl.nasa.gov/perl/agents/interface_agent.pl) is implemented as a Web service using the RQDL (RDF Query Language). This engine can locate NASA data by finding related terms instead of having an exact keyword match. These terms may be synonymous (same as), more specific (subclass), or less specific (superclass) than those requested. Once the synonyms and subclass-superclass relationships have been discovered, the engine then submits the union of these terms to the GCMD search tool. The search results are presented as GCMD DIF summaries.

Geospatial Web Service Composition

A geospatial Web service is a modular application designed to deal with geospatial information over the network. In order to solve real-world geospatial problems through Web services, an "intelligent" mechanism is required to facilitate service discovery and integration and to automate the assembly of service chains. In [18], an approach to intelligent geospatial Web service is presented. The approach uses an ontologized "Geo-Object", a component of "Geo-Tree", to integrate the views of geospatial services and make them understandable and inferable. By using a semantic-enabled OGC catalog service, a "Geo-Object" can be easily found based on the flexible semantic match on its inputs and outputs. In fact, a "Geo-Tree" describing a geospatial modeling process can be represented by a service chain. The automatic construction of such a tree can be conducted through backward reasoning (from goal to source) in which all relevant "Geo-Objects" are discovered automatically based on a user's goals. Once all components have been found, the tree is initiated as a service chain in BPEL (Business Process Execution Language) and then sent to the workflow engine for execution.

Future Directions

From representation logic to computational logic, the GSW enhances our ability to express and deduce geospatial concepts and relationships for achieving interoperability among distributed heterogeneous geospatial data and applications. To make the GSW more efficient, practical, and operational, researchers and developers are investigating the following area [5,19]:

- Mappings between semantics and structured data files and databases, e. g., how can an ontology be generated automatically from maps, images and sketch available on the Web
- Ontology integration, e. g., how to assess computational models to specify, represent, access, and share multiple ontologies of geographic information
- Inference engines, e. g., how do users query the GSW using natural language
- Issues of trust, e. g., how to determine ontology reliability.
- Issues of proofs, e. g., what inferences lead to this answer and are the supporting ontologies logically sound.

Cross References

► Geospatial Semantic Web
► OGC's Open Standards for Geospatial Interoperability
► Ontology-Based Geospatial Data Integration

Recommended Reading

1. Sheth, A.: Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics. In: Goodchild, M., Egenhofer, M., Fegeas, R., Kottman, C. (eds.) Interoperating Geographic Information Systems, pp. 5–30. Kluver, New York (1999)

2. Berners-Lee, T. (1998) Semantic Web Road Map. http://www.w3.org/DesignIssues/Semantic.html. Accessed 20 Mar 2007
3. Egenhofer, M. (2002) Toward the Semantic Geospatial Web. In: The Tenth ACM International Symposium on Advances in Geographic Information Systems. Available via. http://www.spatial.maine.edu/~max/SGW02.pdf. Accessed 8-9 Nov 2002
4. Lieberman, J., Pehle, T., Dean, M. (2005) Semantic Evolution of Geospatial Web Services. In: W3C Workshop on Frameworks for Semantic in Web Services. Available via. http://www.w3.org/2005/04/FSWS/Submissions/48/GSWS_Position_Paper.html. Accessed 9-10 June 2005
5. Fonseca, F., Sheth, A. (2002) The Geospatial Semantic Web. University Consortium for Geographic Information Science. http://www.ucgis.org/priorities/research/2002researchPDF/shortterm/e_geosemantic_web.pdf. Accessed 21 March 2007
6. W3C Semantic Web. http://www.w3.org/2001/sw/. Accessed 21 March 2007
7. Berners-Lee, T. (2000) Semantic Web on XML. In: XML 2000. Available via. http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide1-0.html. Accessed 21 March 2007
8. Antoniou, G., Harmelen, F.: A Semantic Web Primer. MIT Press, Cambridge, MA (2004)
9. Gruber, T.R.: A Translation Approach to Portable Ontologies. Knowledge Acquisition **5**, 199–220 (1993)
10. Chandrasekaran, B., Johnson, T., Benjamins, V.: Ontologies: what are they? Why do we need them?. IEEE Intelligent Syst. Their Appl. **14**, 20–26 (1999)
11. Arpinar, A., Sheth, A., Ramakrishnan, C.: Geospatial Ontology Development and Semantic Analytics. In: Wilson, J.P., Fotheringham, A.S. (eds.) Handbook of Geographic Information Science, Blackwell Publishing, Boston (2004)
12. DublinCore Dublin Core Metadata Initiative. http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide1-0.html. Accessed 21 March 2007
13. Cycorp OpenCyc. http://www.cyc.com/opencyc. Accessed 21 March 2007
14. Brickley, D. (2003) Basic Geo (WGS84 lat/long) Vocabulary. http://www.w3.org/2003/01/geo/. Accessed 21 March 2007
15. Chen, H., Finin, T., Joshi, A. (2003) An Intelligent Broker for Context-Aware Systems. In: Ubicomp (2003), Seattle, Washington, USA. Available via. http://ebiquity.umbc.edu/paper/html/id/78/An-Intelligent-Broker-for-Context-Aware-Systems. Accessed 12–15 Oct 2003
16. Raskin, R.: Guide to SWEET Ontologies. http://sweet.jpl.nasa.gov/guide.doc. Accessed 21 March 2007
17. Islam, A.S., Bermudez, L., Beran, B., Fellah, S., Piasecki, M.: Ontology for Geographic Information – Metadata (ISO 19115:2003). http://loki.cae.drexel.edu/~wbs/ontology/iso-19115.htm. Accessed 21 March 2007
18. Di, L., Zhao, P., Yang, W., Yu, G., Yue, P.: Intelligent Geospatial Web Services. In: Proceedings of IEEE 25th Anniversary International Geoscience and Remote Sensing Symposium. Seoul, Korea (2005)
19. Lowe, J.A.: Geospatial Semantic Web. http://www.geospatial-online.com/geospatialsolutions/article/articleDetail.jsp?id=163883. Accessed 21 March 2007
20. Lieberman, J., Lutz, M., Pehle, T., Kolas, D., Morris C., Dean M., Probst, F., Klien, E.: Geospatial Semantic Web: An Interoperability Experience. http://www.semantic-conference.com/Presentations_PDF/Lieberman-Joshua.pdf (2007). Accessed 31 Aug 2007
21. Lieberman, J.: Geospatial Semantic Web Interoperability Experiment Report.http://portal.opengeospatial.org/files/?artifact_id=15198 Accessed 22 March 2007

Geospatial Semantic Web: Personalization

PRAGYA AGARWAL
Department of Geomatic Engineering,
University College London, London, UK

G

Synonyms

Adaptation; Embodiment, Individualization; Cognitive engineering; Externalization; Reasoning; User-centered; Contextualization; Personalization

Definition

What is the Semantic Web?

The basic principle underlying the semantic web is the idea of having data defined and linked in such a way that it can be used for more effective discovery, automation, integration, and reuse across various applications [7,16]. With the idea of a semantic web in which machines can understand, process and reason about resources to provide better and more comfortable support for humans in interacting with the World Wide Web, the question of personalizing the interaction with web content to user needs and profile is of prime significance.

What is the Geospatial Semantic Web?

Geographic data has meanings associated with it and semantic annotations of geographic data will allow for better retrieval and more effective use of geospatial services. Recent developments in the semantic web have great potential for the geospatial community, specifically because the focus on the incorporation of data semantics will lead to a better retrieval and more reliable integration methods by tapping into the semantics during the search process on the web. There is a distinct move away from structure and syntax in the geospatial community, accompanied by an increased awareness that semantics is the backbone for a successful ontology to enable translation of data from different resources and users. Ontologies are being increasingly used in the geospatial community to enable interoperability between multiple resources, systems, services and semantic conceptualizations. The most commonly used definition of an ontology is that it is a "specification of a conceptualization" [18]. However, the basic semantic web and the technological developments are not targeted to the needs of the geospatial community as spatial data has its own specific needs associated with it. A geospatial domain is characterized by vagueness, especially in the semantic disambiguation of the concepts in the domain, which makes defining universally accepted geoontology an onerous task [2]. This is compounded by the lack of appropriate methods and techniques where the individual semantic conceptualizations can be captured and compared to each other. The idea of a more focused "Geospatial Semantic Web" has been recognized as a research priority within UCGIS (University Consortium for Geographic Information Science) initiatives [17]. Egenhofer [15] identified the need to support queries based on meanings and better definition of spatial terms across a number of disciplines, and the need to integrate multiple terminological ontologies as a backbone for an effective geospatial semantic web (GSW).

What is Personalization?

Nowadays, most large-scale applications are planned and designed for a large variety of end users. Nevertheless, the traditional "one-size-fits-all" approach turns out to be outdated and inappropriate to meet with heterogeneous user needs. "Personalization" is the key word and the specific challenge is to find out what the users are looking for in order to understand how to offer them tailored information. Therefore, personalization of information means to deal with information more efficiently and effectively, performing complex tasks within a reasonable time, in a "user-friendly" environment. The development of any personalization techniques and adaptive information retrieval systems should deal with the users' understanding of domain examined by knowledge capture approaches.

A "closed world" paradigm in which individual preferences and goals are not considered, in which the system requirements are deterministic, and not adapted to changing needs of the community, has meant that the system design and operation work on a fixed set of resources which are normally known to the system designers at design time. On the other hand, personalization facilitates a move away from this deterministic approach that traditional information system design frameworks operate under. The individual requirements and context that the user operates under has to be taken into account to move from a closed-world setting. knowLedge-enhanced web services are normally driven by some description of the world which is encoded in the system in the form of an ontology defined by knowledge engineers. The users' conceptualization of the world may also differ, sometimes significantly, from the conceptualization encoded in the system. If not taken into account, the discrepancies between a user's and a system's conceptualization may lead to the user's confusion and frustration when utilizing semantic web services, which, in turn, can make these services less popular. Semantic personalisation of web-based services is required to exploit user intentions and perspectives and tailor the information accordingly.

Personalization can be defined as a process of gathering user-information during interaction with the user, which is then used to deliver appropriate content and services, tailored to the user's needs. Personalization is a process by which it is possible to give the user optimal support in accessing, retrieving, and storing information, where solutions are built so as to fit the preferences, the characteristics and the taste of the individual. Personalization of information systems aims at developing systems that can autonomously interoperate—either with humans or with other systems—adapting their processing and its outcome to specific requests. Personalized information systems aim to make user needs the center of the interaction process and to optimize user access to the relevant and required information according to individual needs, profiles and particular contexts. This result can be achieved only by exploiting machine-interpretable semantic information, e. g., about the possible resources, about the user him/herself, about the context, about the goal of the interaction. Personalization is realized by a reasoning process applied to the semantic information, which can be carried out in many different ways depending on the specific task.

Although, conventionally Personalization is taken to mean directing the system design to user needs and profiles, it can also mean to adapt according to local browser or regional contexts. The individual requirements of the user are to be taken into account in such different dimensions. Some of these individual user requirements will include the current task, the goal of the user, the context in which the user is requesting the information, the previous information requests or interactions of the user, the working process s/he is involved in, knowledge of the user (an expert will be satisfied by information which is not suitable for a layman), the device s/he is using to display the information, the bandwidth and availability of the communication channel, the abilities/disabilities/handicaps of the user, and the time constraint of the user (whether s/he is under time pressure, or is just browsing some information).

Personalization of Semantic Web

Personalized semantic web is the concept of the semantic web in which machines are enabled to understand the meaning of information and thereby provide better support to individuals in carrying out their tasks, and is aimed at improving the user's experience of a web-based service. In particular, applications that can retrieve, process and present information in enhanced user-adapted ways are interesting.

Many attempts have been made to apply personalization techniques to the World Wide Web as a natural extension of work on hypertext and hypermedia; however, the web is an information space thought for human to human communication, while personalization requires software systems to take part to the interaction and help [13]. Such systems require knowledge to be expressed in a machine-interpretable format, which in the web is not available. The development of languages for expressing information in a machine-processable form is characteristic of the semantic web initiative [6]. Over this knowledge layer, the use of inferencing mechanisms is envisioned as a fundamental means for performing a content-aware navigation, producing an overall behavior that is closer to the user's intuition and desire. This is the reason that the semantic web is the most appropriate environment for realizing personalization. In other words, the semantic web is deeply connected to the idea of personalization in its very nature.

What is Personalization of GSW?

Users' preferences, expectations, goals and tasks differ while using the web for geographic information. Moreover, people form different conceptual models of the world and these models dynamically change over time. In addition, meanings are crucial in distinction of geographic information and people constantly assign meanings to real world objects, while categorizing them as they interact with the world around them. Agarwal [2] discusses in detail the problems associated with ontology development in the geospatial domain primarily due to semantic ambiguities. The knowledge-enhanced web services are normally driven by some description of the world which is encoded in the system in the form of an ontology defined by knowledge engineers. The user's conceptualization of the world may differ, sometimes significantly, from the conceptualization encoded in the system. If not taken into account, the discrepancies between a user's and a system's conceptualization may lead to the user's confusion and frustration when utilizing the web-based geospatial services, which, in turn, can make these services less popular. Indeterminacy and ambiguity in meanings are key issues in the development of ontologies in the geographic domain [2,5]. Empirical results show that individual conceptualizations are characterized by semantic heterogeneity (Agarwal [1]).

With multiple user conceptualizations, efforts towards a reliable geospatial semantic web, therefore, require personalization where user diversity can be incorporated and targeted to multiple needs, semantics and conceptualizations of the real world. Egenhofer [15] identified the need to support queries based on meanings and better definition of spatial terms across a number of disciplines, and the need to integrate multiple terminological ontologies as a backbone for an effective GSW. The success of a standardized geoontology for the semantic web will be determined by the level of acceptance by the users of the services, both expert and naïve, and the level to which the basic geoontology is semantically compatible with the users' conceptualizations.

In this way, personalization is crucial for acknowledging and addressing individual user differences and needs and providing user-friendly user access to geospatial information on the web. To achieve a GSW will require both syntactic and semantic interoperability of resources, and therefore the personalization efforts are essential in making the GSW a reality.

G

Historical Background

GSW is a new initiative and many concrete applications of it are yet to be seen. Personalization of GSW is a novel concept and has no historical precedents.

Scientific Fundamentals

Many research disciplines have contributed to web personalization research, for example, hypertext research has studied personalization in the area of so-called adaptive hypertext systems, collaborative filtering research has investigated recommender systems, artificial intelligence techniques have been widely used to cluster web data, usage data, and user data, reasoning and uncertainty management has been adopted to draw conclusions on appropriate system behavior, etc. [13]. However, in most systems, there are no mechanisms to capture the interaction and context of the user. There is an urgent need to include the people as an axis in the design, development, and deployment of semantically enriched services, especially for personalization of the GSW where individual user semantics vary. In addition, computational models are needed that can process the different terminological and semantic ontologies and process the semantic incompatibilities between users and the expert's geoontology.

Resolving the discrepancy between psychological user variables and physical system variables in the area of geospatial services goes beyond the user-interface level. Rather than a closed view of the world, the personalization efforts for geospatial services design will ensure that the different perspectives and semantic conceptualizations of the real world are maintained as "open". The underlying principle for the methodology adopted by Agarwal et al. [3] and Huang et al. [20] is that such geospatial services and applications could benefit from personalization efforts where semantic variations and conceptualizations are captured and aligned, and differences with the core expert ontology identified and formalized to provide the basis for user-supported access to such location-based information. This will allow, first, for the development of systems that allow personalization by incorporating user models and diversity and second, as a means to test any core ontologies that are developed as the basis for a geospatial services against user conceptualizations for discrepancies and thereby evaluate its reliability as a standard, reusable ontology. Moreover, the personalization approach allows flexibility and the possibility of using the user models to enrich the available information resources with shared semantics instead of relying on fixed ontologies available to the developers at the design stage. Using this approach, the research towards specification of well-defined standards and ontologies for interoperability in the geospatial domain can be enhanced and personalized to provide extendibility, flexibility, interoperability, and reusability. Automating the mapping of multiple conceptualizations and personalization of web-based services will also facilitate pervasive computing in mobile services and enable more effective use of mobile GIS services.

Key Applications

The availability of geospatial knowledge resources on the web enables members of the public to take advantage of trusted knowledge built by domain experts, e. g., for planning travel routes and for accessing weather information. Geospatial services and systems are also unique in the way that they use data, which are related to locations in space and time, and that the processing of the data with respect to these spatial locations is possible. People's input to geospatial tools have a spatiotemporal context. One can ask "where is a certain object" or "where are all objects with certain properties" at a given time when trying to find the nearest health services for the elderly; or one can ask "what are the properties of a certain area in space (as well as time)" when trying to ascertain the suitability of an environment (for example, crime rates) while renting or buying a property. Users access the web services with different goals and often; these services require integration of the various different resources to provide a comprehensive result for the user search for their specific requirements. For example, in a "what is in my backyard" service provided by the Environment Agency (EA) in the UK, members of the public can see what pollutants may be scattered across their neighborhood. End-users will have their own contexts of use: property evaluation, ecology, etc. and for a member of the public, a general interest (based on a topographic view of different areas in the city). Each could potentially view the data provided by the others but form their own conceptual understanding of the location-based information. Personalization efforts will make it possible to capture and formalize context and thereby provide context-relevant information.

Future Directions

There is no common language for expressing adaptive functionality, hence these systems are difficult to compare and analyze, also suffering to a great extent from lack of reusability—which in fact is the key capability for successful personalization functionality for the GSW. To deal with a diverse user population having different preferences, goals, understanding of tasks and conceptual models, existing design paradigms in geospatial services will have to be redefined. Furthermore, new diagnostic techniques and models are needed to capture the long-term development of users' capabilities, the dynamics of users' goals and conceptual understanding, the uncertainty and inconsistency of naive users' conceptualizations, and so on. The ambitious target is to offer manageable, extendible and standardized infrastructure for complementing and collaborating applications tailored to the needs of individual users. Traditional personalization and adaptation architectures were suited to deal with closed-world assumption, where user modeling methods, such as overlay, bug library, constraint-based modeling and other marked discrepancies in a user and expert's semantics as erroneous, and often called them misconceptions. New approaches for open-world user modeling that facilitate elicitation of extended models of users are needed to deal with the dynamics of a user's conceptualization. Similarly, methods that acknowledge semantic discrepancies and heterogeneity are required for effectively personalizing the web-based services for the geospatial community. Adaptation and personalization methods are not developed to address the temporality inherent in geospatial information and therefore, any specific personalization efforts for the GSW will have to be modified to suit this need. Without the benefit of deeper semantic or ontological knowledge about the underlying domain, personalization systems cannot handle heterogeneous and complex objects based on their properties and relationships. Nor can these systems possess the ability to automatically explain or reason about the user models or user recommendations. This realization points to an important research focus that can combine the strengths of web mining with semantic or ontological knowledge. Development of automated reasoning tools to detect mismatches and discrepancies between the user and the expert ontology forming the backbone for the web-based resources will be a step forward.

Cross References

► Geospatial Semantic Web

Recommended Reading

1. Agarwal, P.: Contested Nature of place. In: Egenhofer, M., Freska, C., Harvey, M. (eds.) 3rd International Conference GIScience, 2004, LNCS 3234, pp. 1–21. Springer, Berlin (2004)
2. Agarwal, P.: Ontological considerations in GIScience. Int. J. Geogr. Inf. Sci. **19**, 501–535 (2005)
3. Agarwal, P., Huang, Y., Dimitrova, V.:, Formal approach to reconciliation of individual ontologies for personalization of geospatial semantic web. In: Proceedings of the First International Conference on Geospatial Semantics, Mexico City, November 2005. Lecture Notes in Computer Science, vol. 3799, pp.202–219, Springer-Verlag, Berlin (2005)
4. Antoniou, G., van Harmelen, F.: A Semantic Web Primer. MIT Press, Cambridge, Massachusetts (2004)
5. Bennett, B.: What is a Forest? On the vagueness of certain geographic concepts. Topoi **20**, 189–201 (2001)
6. Berners-Lee, T.: (1998) The Semantic Web Road Map. World Wide Web Consortium W3C Draft. http://www.w3.org/DesignIssues/Semantic
7. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Sci. Am. May (2001)
8. Brusilovsky, P.: Adaptive Hypermedia. User Modeling User-Adapted Interaction **11**, 87–110 (2001)
9. Cardoso, J.: Semantic Web Services: Theory, Tools and Applications. Idea Group, (2007)
10. Cardoso, J., Sheth, A.: Semantic Web Services, Processes and Applications. Springer-Verlag, USA (2006)
11. Davies, J., Fensel, D., van Harmelen, F.: Towards the Semantic Web: Ontology-Driven Knowledge Management. Wiley, UK (2003)
12. Doan, A., Madhavan, J., Domingos, P., Halvey, A.: Learning to map between ontologies on the semantic web. In: Proceedings of WWW'02, Hanau, 7–11 May 2002
13. Dolog, P., Henze, N., Nejdl, W., Sintek, M.: (2003) Towards the adaptive semantic web. In: Proceedings of the 1st Workshop on Principles and Practice of Semantic Web Reasoning (PPSWR'03), co-located with International Conference on Logic Programming (ICLP'03), Mumbai, India, December 8, 2003
14. Dolog, P., Henze, N., Nejdl, W., Sintek, M.: Personalization in Distributed e-Learning Environments. In: Proceedings of 13th World Wide Web Conference, New York, USA 17–22 May 2004
15. Egenhofer, M.J.: Toward the semantic geospatial web. In: Proceedings of the Tenth ACM International Symposium on Advances in Geographic Information Systems, McLean, Virginia, 8–9 Nov 2002
16. Fensel, D., Wahlster, W., Lieberman, L., Hendler, J.: Spinning the Semantic Web: Bringing the World Wide Web to its Full Potential. MIT Press, Cambridge, MA (2002)
17. Fonseca, F., Sheth, A.: (2002) The geospatial semantic web. UCGIS White Paper. http://www.ucgis4.org/priorities/research/2002researchagenda.htm
18. Gruber, T.R.: A translation approach to portable ontologies. Knowledge Acquisition **5**, 199–220 (1993)
19. Hendler, J., Berners-Lee, T., Miller, E.: Integrating applications on the semantic web (English version). J. Inst. Electrical Eng. Jpn. **122**, 676–680 (2002)
20. Huang, Y., Dimitrova, V., Agarwal, P.: Detecting mismatches between a user's and a system's conceptualizations to empower user modelling for the Semantic Web. In: Proceedings of the Workshop on Personalization in the Semantic Web, User Modelling Conference 2005, Edinburgh, 24–30 July 2005
21. Shadbolt, N., Berners-Lee, T., Hall, W.: The semantic web revisited. IEEE Intell. Syst. **21**, 96–101 (2006)
22. Staab, S., Studer, R.: Handbook on Ontologies. Springer-Verlag, Heidelberg (2004)

Geospatial Semantics

► Geospatial Semantic Integration

Geospatial Semantics Web

► Geospatial Semantic Web: Applications

Geospatial Technology

► Evolution of Earth Observation

Geospatial Web 3.0

► Geospatial Semantic Web: Applications

Geostatistical Models

► Hierarchical Spatial Models

Geostatistics

► Data Analysis, Spatial

Geotemporal Role-Based Access Control

► Security Models, Geospatial

Geovista

► Geographic Dynamics, Visualization And Modeling

Geovisualization

► Exploratory Visualization

Getis-Ord Index G*

► Local and Global Spatial Statistics

GIbb's Sampling

► Hurricane Wind Fields, Multivariate Modeling

GIS

► Information Services, Geography

GIS Mashups

ILYA ZASLAVSKY
San Diego Supercomputer Center,
University of California San Diego, San Diego, CA, USA

Synonyms

Composite geographic information systems web application; Web application hybrids

Definition

A mashup is an online application or web site that seamlessly combines content from several sources. Geographic information systems (GIS) mashups typically combine spatial data and maps from several web sources to produce composite thematic maps.

Main Text

Mashup is an approach to creating composite dynamic web sites that has quickly gained popularity due to the proliferation of reliable online sources of maps, web feeds, and other content that can be accessed via relatively simple published interfaces (APIs). Online maps and GIS services available through Google, Windows Live, and Yahoo APIs, among others, can be combined with real time traffic information, weather information, various third party databases, blogs, information feeds, search engines, forums, etc. Mashups are also thought of as "Web 2.0 applications", a somewhat hyped notion of a next generation internet application platform that emphasizes openness of data and services, lightweight application interfaces and the ability to mix and match different services to create a rich application experience for web users.

Cross References

► Cyberinfrastructure for Spatial Data Integration
► Grid
► Web Services

GIS-Based Hydrology

► Distributed Hydrologic Modeling

GIS-Based Multicriteria Decision Analysis

► Multicriteria Decision Making, Spatial

GIS Software

► ArcGIS: General Purpose GIS Software System

GIService

► Information Services, Geography

GiST Index

► PostGIS

Global and Local Spatial Modeling

► Spatial and Geographically Weighted Regression

Global Positioning System

► Location-Aware Technologies

Global Sensitivity Analysis

XIAOBO ZHOU, HENRY LIN, HENRY LIN
Department of Crop and Soil Sciences, The Pennsylvania State University, University Park, PA, USA

Definition

Global sensitivity analysis is the process of apportioning the uncertainty in outputs to the uncertainty in each input factor over their entire range of interest. A sensitivity analysis is considered to be global when all the input factors are varied simultaneously and the sensitivity is evaluated over the entire range of each input factor.

Main Text

In many complex and nonlinear spatial phenomenon or processes, input factors usually interact with each other and therefore it is inappropriate to evaluate the impact of one input factor on the model output with other factors being constant. Global sensitivity analysis quantifies the importance of model inputs and their interactions with respect to model output. It provides an overall view on the influence of inputs on outputs as opposed to a local view of partial derivatives as in local sensitivity analysis. One of the most challenging issues for global sensitivity analysis is the intensive computational demand for assessing the impact of probabilistic variations. There are many global sensitivity analysis methods including the Sobol's sensitivity estimates, the Fourier amplitude sensitivity test (FAST), and the Monte-Carlo-based regression–correlation indices. The global sensitivity analysis is suitable for a nonlinear input–output relationship, and is more realistic to the real world since it allows all input factors to be varied simultaneously.

Cross References

► Local Sensitivity Analysis
► Screening Method
► Sensitivity Analysis

GML

► deegree Free Software
► Geography Markup Language (GML)
► Web Feature Service (WFS)

GNU

► deegree Free Software

Gnu Public License (GPL)

► Open-Source GIS Libraries

Google Earth

► Multimedia Atlas Information Systems

G-Polygon

► Spatial Data Transfer Standard (SDTS)

GPS

► Evolution of Earth Observation
► Indoor Localization
► Location-Aware Technologies
► Moving Object Uncertainty
► Privacy Preservation of GPS Traces

GPSBabel

► Open-Source GIS Libraries

Graph

► Road Network Data Model

Graph Theory, Konigsberg Problem

BETSY GEORGE
Department of Computer Science,
University of Minnesota, Minneapolis, MN, USA

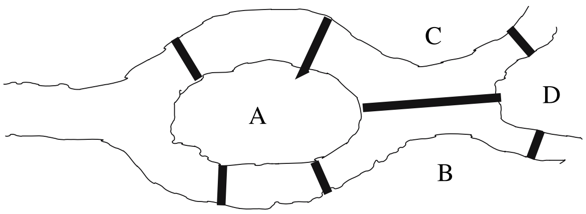
Synonyms

Problem of seven bridges of Konigsberg; Euler's Konigsberg's bridges problem; Edge routing problems

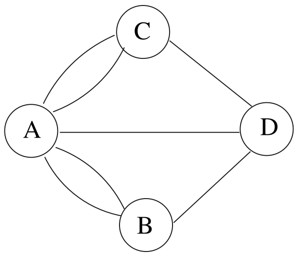
Definition

In Geographic Information Systems, concepts from graph theory are extremely useful in expressing the spatial structure of entities seen as points, lines, areas and solids, after the geometrical details of these entities are removed. For example, in transportation and river networks, the topological properties of their structures can be represented using graphs. This article describes the origins of graph theory and the impact it has on various fields ranging from geography to economics.

The Konigsberg Bridge Problem is a classic problem, based on the topography of the city of Konigsberg, formerly in Germany but now known as Kalingrad and part of Russia. The river Pregel divides the city into two islands and two banks as shown in Fig 1. The city had seven



Graph Theory, Konigsberg Problem, Figure 1 Layout of the city of Konigsberg showing the river, bridges, land areas



Graph Theory, Konigsberg Problem, Figure 2 Graph Representation ot the city of Konigsberg

bridges connecting the mainland and the islands (represented by thick lines in the figure). [1,2,3,4]. The problem asks whether there is a walk that starts at any island, traverses every bridge exactly once, and returns to the start point.

The solution proposed by a Swiss Mathematician, Leonhard Euler, led to the birth of a branch of mathematics called graph theory which finds applications in areas ranging from engineering to the social sciences. Euler proved that there is no solution to the problem based on the number of bridges connecting each land area.

The results from the solution of the Konigsberg problem have been extended to various concepts in graph theory. In graph theory a path that starts and ends at the same node and traverses every edge exactly once is called an Eulerian circuit. The result obtained in the Konigsberg bridge problem has been generalized as Euler's theorem, which states that a graph has an Eulerian circuit if and only if there are no nodes of odd degree. Since the graph corresponding to Konigsberg has four nodes of odd degree, it cannot have an Eulerian circuit. Subsequently the concept of Eulerian paths was introduced, which deals with paths that traverse every edge exactly once. It was proved that such a path exists in a graph if and only if the number of nodes of odd degree is 2 [3,4,5,8,9].

While studying the Konigsberg bridge problem, Euler also observed that the number of bridges at every land area would add up to twice the number of bridges. This result came to be known as the hand-shaking lemma in graph theory, which states that the sum of node-degrees in a graph is equal to twice the number of edges. This result is the first formulation of a frequently used result in graph theory that states that the sum of node-degrees in a graph is always even [8,9].

Historical Background

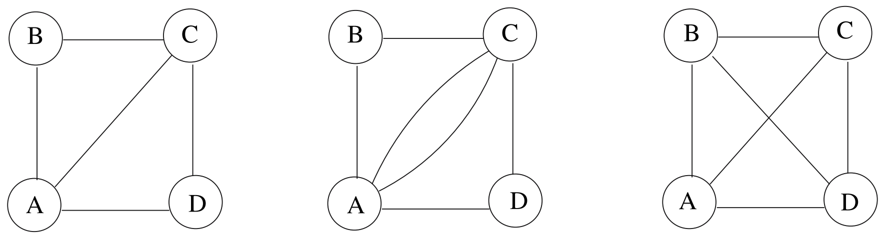
The Konigsberg bridge problem was formulated based on the layout of the city of Konigsberg around the river Pregel. The problem was to find a tour that starts at any point in the city, crosses each bridge exactly once, and returns to the starting point. No one succeeded in doing this. A Swiss mathematician, Leonhard Euler formulated this problem by abstracting the scenario. He formulated the problem as finding a sequence of letters A, B, C, D (that represent the land areas) such that the pairs (A,B) and (A,C) appear twice (thus representing the two bridges between A and B, and A and C) and the pairs (A,D), (B,D), (C,D) appear only once (these pairs would represent the bridges between A and D, B and D, C and D). Euler used a counting argument to prove that no such sequence exists thus proving that there the Konigsberg Bridge Problem has no solution. Euler presented this result in the paper, "The Solution of Problem Relating to the Geometry of Position" at the Academy of Sciences of St. Petersburg in 1735. This paper, in addition to proving the non-existence of solution to the Konigsberg Bridge Problem, gave some general insights into arrangements of bridges and land areas [5,6,8].

Euler summarized his main conclusions as follows:

1) If there is any land area that is connected by an odd number of bridges, then a cyclic journey that crosses each bridge exactly once is impossible.
2) If the number of bridges is odd for exactly two land areas, then there is a journey that crosses each bridge exactly once is possible, if it starts at one of these areas and ends in the other.
3) If there are no land areas that are connected by odd number of bridges, the journey can start and end at any land area [8].

Euler gave heuristic reasons for the correctness of the first conclusion. To complete a cyclic journey around the land areas, crossing each bridge exactly once, there must be a bridge to leave the area for every bridge to enter it. This argument was generalized to the conclusion that a cyclic journey is possible if every island is connected by an even number of bridges. Formal proofs for the conclusions were not proposed until the year 1871, in a posthumous paper by Carl Hierholzer [2,5].

The paper presented by Euler on the Konigsberg bridge problem can be considered to mark the birth of graph theory, in general. Later, a diagrammatic representation evolved which involved nodes or vertices and the connecting lines that are called edges. Using this representation, the Konigsberg problem is modeled as shown in Fig. 2.



a
Eulerian Path:
A–B–C–D–A–C

b
Eulerian Cycle:
A–B–C–D–A–C–A

c
Neither eulerian path
nor cycle exist

Graph Theory, Konigsberg Problem, Figure 3 Illustration of an eulerian path and eulerian cycle

Circles, called nodes, represent the islands and the banks and connecting lines called edges represent the bridges. The number of edges that are incident on a node is called the degree of the node [8].
In the Konigsberg bridge problem, the number of bridges connecting a land area would be the degree of the node representing the land area.

Scientific Fundamentals

In an undirected graph, a cycle that traverses every edge exactly once is called an Euler tour or Euler cycle. Any graph that possesses an Euler cycle is called an eulerian graph. A path that traverses each edge exactly once with different starting point and end point is called an eulerian path. An undirected multi-graph has an Eulerian circuit (path) if and only if it is connected and the number of vertices with odd degree is zero (two).
Figure 3 illustrates eulerian path and eulerian cycle in a graph. In Fig. 3a, an eulerian path exists and it can be observed that the graph has exactly two, odd degree vertices, which would be the start and end vertices of the eulerian path, A-B-C-D-A-C. Figure 3b does not have vertices with odd degree and has an eulerian cycle whereas Fig. 3c has neither an eulerian path nor an eulerian cycle.

Finding an Eulerian Circuit in a Graph

The method successively finds cycles in the graph. At each step the edges that are in the already discovered cycles are removed and the cycle is spliced with the one discovered in the previous step. This process is continued until all edges are exhausted. These basic ideas were formalized into an algorithm in [10]. The algorithm maintains a list L with each vertex x such that the kth entry in the list indicates the vertex to visit when vertex x is reached the kth time.

Algorithm

Step 1: Select any vertex $v1$. $v = v1$; $k[v] = 0$ for all vertices v. Label all edges as unvisited.

Step 2: Select an unvisited edge e incident to v. Mark this edge "visited". Let w be the other end vertex of e. Increment k_v by 1 and $L_v[k_v] = w$. If w has an unvisited incident edge, go to step 3. If not, y will be $v1$. Then, go to step 4.

Step 3: Set $v = w$ and go to step 2.

Step 4: Find a vertex $v1$ such that there is at least one visited edge and one unvisited edge incident at $v1$. Set $v = v1$ and go to step 2. If no such vertex exists, go to step 5.

Step 5: To construct the Eulerian circuit, start at $v1$. First time a vertex u is reached, proceed to the vertex $L_u[k_u]$. Decrement k_u and continue.

Trace of the algorithm for Figure 4

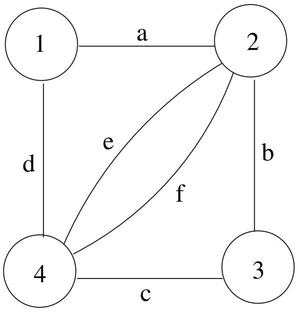
Step 1: $v1 = 1 = v$; $k_x = 0$ for $x = 1, 2, 3, 4$.
Step 2: Select edge a. $w = 2$; $k_2 = 1$; visited$(a) = 1$.
Step 3: $v = 2$; Select edge b. $w = 3$; $k_3 = 1$; visited$(b) = 1$.
Step 4: $v = 3$; Select edge c. $w = 4$; $k_4 = 1$; visited$(c) = 1$.
Step 5: $v = 4$; Select edge d. $w = 1$; $k_1 = 1$; visited$(d) = 1$.
Step 6: $v = 2$;
Step 7: Select edge e; $w = 4$; $k_4 = 2$; visited$(e) = 1$
Step 8: $v = 4$;
Step 9: Select edge f; $w = 2$; $k_2 = 2$; visited$(f) = 1$
Step 10: Construct the cycle as 1–2–4–2–3–4–1.

Key Applications

Eulerian cycles find applications in problems where paths or cycles need to be found that traverse a set of edges in a graph. Such problems are generally called edge routing problems.

Snow Plow Problem

This problem requires finding the least distance route in the road network that starts and ends at the station so that snow



Graph Theory, Konigsberg Problem, Figure 4 Illustration of eulerian algorithm

can be cleared from the streets at the minimum cost. The minimum distance route is obviously the eulerian cycle since this cycle traverses each edge exactly once. However, it is unlikely that any real road network would happen to satisfy the necessary conditions that make it Eulerian. In that case, the problem moves to the realm of "the Chinese postman problem" [10,11,12].

Chinese Postman Problem

A postman delivers mail everyday in a network of streets. It is useful to know whether or not, the postman can traverse the network and return to the mail station without driving the length of any street more than once. If the network is not eulerian, the problem gets modified to the one where it is required to find the shortest path, which visits each edge at least once. This problem statement requires a parameter to be associated with each edge that represents the cost of traversing that edge. For example, cost can be the represented in terms of the length of the street, which the edge represents.

In a non-eulerian graph, the postman's circuit, shortest or otherwise, will repeat one or more edges. Every vertex is entered the same number of times that it is left so that any vertex of odd degree has at least one incident edge that is traversed at least twice. Chinese postman problem is formulated as an optimization problem where the total cost of repeated edges in minimized.

Algorithm

Step 1: Find the shortest path between each pair of odd-degree.

Step 2: Find the sub-graph G' with the odd degree vertices.

Step 3: Find the minimum weight matching of all the edges in G'. The edges in the shortest path connecting a matched pair of odd degree vertices should be repeated.

Figure 5 shows a sample graph with edge weights and the Chinese postman algorithm finds the least cost (minimum edge weight) path in the graph such that every edge is traversed at least once. Table 1 shows the shortest path costs between every pair of vertices, which is used by the algorithm to find the minimum weight matchings on edges (the matchings and their costs are listed in Table 2). The algorithm finds that the paths from vertex 1 to vertex 3, and the path from 2 to 4 must be repeated since this is the minimum cost matching (cost is 5). The algorithm finds the optimal route to be 1–2–3–4–2–4–1–3–1 in the graph shown in Fig. 5.

Capacitated Chinese Postman Problem

This problem arises where each edge has a demand and vehicles to be routed have finite capacities. For example, in applications involving road salting in winter season, there is a limit on the maximum amount of salt that a truck can carry. The amount of salt required is fixed for a road segment. The Capacitated Chinese postman problem finds the sets of routes from a single station that service all the road segments in the network at a minimal cost and subject to the constraint that the total demand of each route does not exceed the capacity of each truck. Christofides' proposed an algorithm to solve this problem.

Capacitated Arc Routing Problem

This problem is different from the Capacitated Chinese postman problem in that demands of some of the road segments can be zero. This situation can arise in road salting scenarios where state highways can be used for traveling, but need not be salted. These edges can be used to traverse between the edges that require the service.

Both the Capacitated Chinese Postman Problem and Capacitated Arc Routing Problem are NP-hard [12] and heuristic methods are normally used to obtain solutions.

Graph Theory

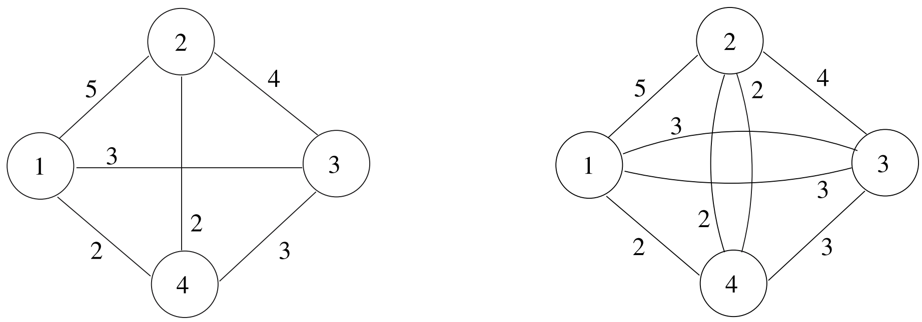
The Konigsberg problem had a powerful impact on mathematics, paving the way for the creation of a new modeling

	1	2	3	4
1	0	4	3	2
2	4	0	4	2
3	3	4	0	3
4	2	2	3	0

Graph Theory, Konigsberg Problem, Table 1 Shortest Path Cost between the pairs

Matching	Cost
(1,2),(3,4)	4+3=7
(1,4),(2,3)	2+4=6
(1,3),(2,4)	3+2=5

Graph Theory, Konigsberg Problem, Table 2 Matchings and Costs



Graph Theory, Konigsberg Problem, Figure 5 Illustration of Chinese postman problem algorithm

theory called graph theory. The applications of graph theory are numerous in science and engineering. A few are listed below

Graph Theory in Spatial Networks The very fact that graph theory was born when Euler solved a problem based on the bridge network of the city of Konigsberg points to the apparent connection between spatial networks (e. g. transportation networks) and graphs. In modeling spatial networks, in addition to nodes and edges, the edges are usually qualified by adding weights that encode information like the length of the road segment that the edge represents. Connectivity and shortest paths in spatial networks have been extensively studied using graphs [13].

Graph Theory in Geography Graphs are also widely applied in geography in the modeling of stream systems. Streams have been modeled as hierarchical graphs and random graphs in the literature [14].
In addition to the applications described above, graphs find other wide applications, including modeling of social networks, molecular structures in Chemistry, computer networks, electrical networks and syntax structures in linguistics.

Future Directions

Relationships between eulerian graphs and other graph properties such as Hamiltonian property are being studied [7]. Graph, the mathematical model which owes its origin to Konigsberg bridge problem, is being increasingly applied to several evolving domains such as spatio-temporal networks, which has necessitated the incorporation of temporal dimension in graphs.

Recommended Reading

1. Chartrand, G.: The Königsberg Bridge Problem: An Introduction to Eulerian Graphs. In: Introductory Graph Theory. Dover, New York (1985)
2. Harary, F.: Graph Theory, pp. 1–2. Addison-Wesley, Reading (1994)
3. Biggs, N.L., Lloyd, E.K., Wilson, R.J.: Graph Theory 1736–1936. Oxford University Press, Oxford (1976)
4. Bollobás, B.: Graph Theory: An Introductory Course, p.~12. Springer, Berlin (1979)
5. Steinhaus, H.: Mathematical Snapshots, 3rd edn. Dover, New York (1999), pp. 256–259
6. Hartsfield, R.: Pearls in Graph Theory. Academic Press, San Diego (1990)
7. Fleischner, H.: (Some of) the many uses of Eulerian Graphs in graph theory, Discrete Math. **230**, 23–43 (2001)
8. Biggs, N.L., Lloyd, E.K., Wilson, R.J.: Graph Theory. Oxford University Press, Oxford (1998)
9. Wilson, R.J.: Introduction to Graph Theory. Longman Inc., New York (1985)
10. Edmonds, J., Johnson, E.L.: Matching, Euler tours and Chinese Postman, Math. Program. **5**, 88–124 (1973)
11. Christofides, N.: The optimum traversal of a graph, Int. J. Manage. Sci. **1**(6) (1973)
12. Evans, J.R., Minieka, E.: Optimization Algorithms for Networks and Graphs. Marcel Dekker Inc., New York (1992)
13. Shekhar, S., Chawla, S.: Spatial Databases: A Tour. Prentice Hall, New Jersey (2003)
14. Wilson, R.J., Beineke, L.W. (eds.): Applications of graph theory. Academic Press, London, New York (1979)

GRASS

Malte Halbey-Martin
Institute for Geographic Sciences, Freie Universität Berlin, Berlin, Germany

Synonyms

Geographic Resources Analysis Support Software; GRASS-GIS; Visualization tool

Definition

GRASS-GIS (Geographic Resources Analysis Support Software) is a powerful open source software for geospatial analysis and modeling that can manage both raster and vector data. In addition it supports three dimensional modeling with 3D raster voxel or 3D vector data and contains several image processing modules to manipulate remote sensing data. It includes visualization tools and interacts with other related software such as the statistical software package R, gstat and Quantum GIS. GRASS supports a wide variety of GIS formats through the use of the GDAL/OGR library. It also supports Open Geospatial Consortium (OGC)—conformal Simple Features and can connect to spatial databases such as PostGIS via ODBC (Open DataBase Connectivity). GRASS datasets can be published on the internet with the UMN Mapserver software.

The software is published under the terms of the GNU General Public Licence (GPL). Anyone can access the source code, the internal structure of the software and the algorithms used. Therefore any user can improve, modify or extend the program according to his own needs. No licence fees have to be paid under the terms of the GPL. Programmers all over the world contribute to GRASS, one of the largest Open Source projects in the world with more than a million lines of source code.

GRASS runs on a variety of platforms including GNU/Linux, MS-Windows, MacOS X and POSIX compliant systems. It is completely written in C although a Java version also exists (JGRASS).

Historical Background

The history of GRASS dates back to the early 1980s when it was developed by the U.S. Army Construction Engineering Research Laboratory (CERL), Champaign, Illinois, USA to meet their needs for land management and environmental planning tools at military installations. Emphasis was placed on raster analysis and image processing because a principal goal was estimation of the impact of actions on continuous surfaces like elevation or soils [27] and there was no adequate raster GIS software on the market at that time. Modules for vector processing were added later.

The first version of GRASS was released in 1984 and because its development was financed by federal funds, US law required the release into the public domain. The complete source code was published on the Internet during the late eighties, a period during which there was significant improvement in its capabilities. CERL withdrew from GRASS development in 1995 and an international team took over this task. In 1997, GRASS 4.2 was published by Baylor University, Waco, Texas, USA. In 1999, GRASS 4.2.1 was released by the Institute of Physical Geography and Landscape Ecology, University of Hannover, Germany. Since GRASS version 4.2.1, GRASS has been published under the terms of the GPL of the Free Software Foundation. In 1999 the work on version 5.0 was started and the headquarters of the "GRASS Developer Team" moved to the Instituto Trentino di Cultura (ITC-irst), Trento, Italy. GRASS 5.0 was released in 2002, followed by version 6.0 in March 2005 which included a complete rewrite of the GRASS vector engine. The current stable version is 6.2 which was released at the end of October 2006.

GRASS is a founding project of the Open Source Geospatial Foundation (OSGeo) which was established in February 2006 to support and build high-quality open source geospatial software.

Scientific Fundamentals

Philosophy of GRASS

The most distinguishing feature of GRASS in comparison to other GIS software is that the source code can be explored without restriction so that anyone can study the algorithms used. This open structure encourages contributions by the user community to the source code in order to improve existing features or to extend it in new directions. For this purpose GRASS provides a GIS library and a free programming manual, which can be downloaded from the GRASS project site [16]. Under the terms of the GPL these contributions can not be distributed in proprietary software unless free access to the source code is granted to the end user. Any published code which is based on GPL licensed code must be licensed under the GPL as well.

GRASS offers the user a wide range of GIS functions. Together with other (free) software tools it provides a complete and powerful GIS software infrastructure at very low cost. GRASS is available on the project's homepage. The design is modular, consisting of more than 350 stand alone modules which are loaded when they are called into a GRASS session.

Interoperability: GIS and Analysis Toolchain

GRASS is designed in a way that offers a high level of robust interoperability with external applications, offering the user tremendous flexibility and efficiency in accomplishing analytic tasks.

Programming and Extending GRASS

GRASS is written in C and comes along with a sophisticated and well documented C/C++ application programming interface (API). As a side benefit of the open source philosophy, the user has the opportunity to learn how to

develop new applications by using existing modules as examples and exploring their source code.
In addition, GRASS functions can be called with high level programming languages like Python using, for example, the GRASS-SWIG (Simplified Wrapper and Interface Generator) which translates ANSI C/C++ declarations into multiple languages (Python, Perl). An integrated parser is provided for scripting languages.
Extensions can be created easily using the extension manager, so that no programming of source code is needed to build additional GRASS modules. Moreover, this modular design helps the user to add new contributions to GRASS without affecting the software suite as a whole.
To automate repeating tasks in GRASS shell scripts can be written.

Relational Database Management Systems

GRASS can connect directly to relational database management systems (RDBMS) such as SQlite, MySQL and PostgreSQL. It supports PostGIS, the spatial extension of PostgreSQL and can connect to other external RDBMS via ODBC interface.

Statistical Analysis

R, a software environment for statistical computing and graphics, can be called within a GRASS session for statistical analysis of geodatasets. Similarly, there is a GRASS interface to gstat, a multivariable geostatistical modeling, prediction and simulation software package. Therefore gstat and R can access both GRASS raster and vector datasets for computations within the spatial region defined in GRASS. This capability creates the potential for both simple and complex geostatistical analysis as shown by [4] and [5]. GRASS can import and export Matlab® binary (.mat) files (version 4) for processing numerical calculations outside GRASS.

Interoperability with Other GIS Software

GRASS supports nearly all common GIS file formats enabling the use of other GIS applications or external datasources. Its binding to the GDAL/OGR library and the support of OGC Simple Features ensure that data exchange between various applications and between multiple users is straightforward. The internal file structure implemented in GRASS, coupled with UNIX-style permissions and file locks, allows concurrent access to any project. Several individuals can share the resources of a single machine and dataset.
GRASS works closely with Quantum GIS. GRASS modules are accessible through a GRASS plugin in Quantum GIS.

2D and 3D Visualization

GRASS comes with fully functional 2D cartography and 3D visualization software (NVIZ). It also interacts with other software tools to enable production of maps or visualization of geographic data sets. GRASS contains an export filter for Generic Mapping Tool (GMT) files and various image formats so that high quality maps for publication can be generated with external image manipulation software.
3D vector and raster datasets can be exported from GRASS as VTK (Visualization ToolKit) files which can be viewed in Paraview, a large data set visualization software package. Script files for Povray, a raytracer to design 3D graphics can be produced, as can VRML (Virtual Reality Modeling Language) files. Animations can be built with NVIZ or the external programs mentioned above.

Web Mapping

The UMN MapServer is able to connect to GRASS and can read GRASS geodatasets directly. With the help of PyWPS (Python Web Processing Service, an implementation of the Web Processing Service standard from the Open Geospatial Consortium) GRASS modules are accessible via web interface and can serve as a backbone in WebGIS applications.

Key Applications

GRASS is currently used around the world in academic and commercial settings as well as by many governmental agencies and environmental consulting companies. Due to the wide variety of applications for spatial data analysis, the following selection gives only a brief overview of situations where GRASS has been adopted. A collection of papers highlighting GRASS implementation can be found here [7].

Archaeology

GIS is of growing importance in this field. In fact GRASS has been widely used in archaeology to support the survey of excavation areas or to simulate ancient processes. GRASS has been used to model the most suitable place to conduct a survey in the Netherlands by [6]. Based on the assumption that settlement actions of ancient people show regional patterns, locations most suitable for archaeologic sites can be deduced. [11] used artificial neural networks as a tool to predict archaeological sites in East Germany and [22] extended GRASS to automate cumulative viewshed analysis. These examples also shows how the potential power of GIS increases when the software is modified by its users for specific needs. Pedestrian hunters and gatherers can be modelled in GRASS using MAGICAL, which consists of three separate GRASS modules [21] to simulate multiagent spatial behavior. [24] proposed how a GRASS based multidimensional GIS framework for archaeological excavations can be developed.

Biology

[31] used GRASS to model the distribution of three bird species in north-east England using a Bayesian rule-based approach. They linked data about habitat preferences and life-history of the birds against physiogeographic and satellite data using GRASS.
On the Iberian Peninsula [15] used GRASS to model the potential area of *Pinus sylvestris*. They predicted the habitat suitability with a machine learning software suite in GRASS incorporating three learning techniques (Tree-based Classification, Neural Networks and Random Forest) in their GIS-based analysis. All three techniques show a larger potential area of *P. sylvestris* than the present model. In the Rocky Mountains National Park tree population parameters have been modeled by [2] for the forest-tundra ecotone.

Environmental Modeling

GRASS is widely used in environmental modeling because of its strong raster and voxel processing capabilities. It offers a variety of techniques to conduct environmental modeling as described in [25]. Besides the use of custom-written models, GRASS includes a large number of models already implemented that can be used for hydrological analysis (Topmodel, SWAT, Storm Water Runoff, CASC2D), watershed calculations and floodplain analysis as well as erosion modeling (ANSWERS, AGNPS 5.0, KINEROS). Models for landscape ecological analysis and wildfire spread simulation also exist within GRASS.

Geography (Human/Physical)

GIS is used in a wide range of analyzes in human and physical geography because both subjects make extensive use of geodata or spatial geodatabases. GRASS is the GIS software of choice in many geographic surveys worldwide.

Geology/Planetary Geology

[20] used a digital elevation model (DEM) and a logical model of the geological structure to derive the surface boundaries of each geologic unit in their study area located in the Izumi mountain range. From these data they built a 3D model of the local geology.

GRASS has also been used in planetary geology. [13] identified Wrinkle Ridges on Mars which can be an evidence of existing subsurface ice on the planet. They used Mars MGS and Viking Mission data to perform their study. The mapping of geologic features from Mars data was done by [10]. The authors detected tectonic surface faults and assigned them to a geologic Mars region. The ability to import the raw data from various Mars datasets and to reproject them quickly and accurately is seen as a great benefit by the authors.

Geomorphology / Geomorphometry

Modules for surface analyzes in GRASS offer the possibility to derive terrain parameters like slope, aspect, pcurve and tcurve in one step. [3] has shown how the geomorphology of a exemplary study area in Kosovo can be statistically analysed with GRASS and R. From a subset of GTOPO30 elevation data he performed various statistic computations on selected relief parameters leading to a classification of geomorphologic units. [17] has used the combination of GRASS and R to perform morphometric analysis of a mountainous terrain in Brazil. With this package he derived morphometric parameters (hypsometry, slope, aspect, swat profiles, lineament and drainage density, surface roughness, isobase and hydraulic gradient) from DEMs and analysed these parameters statistically.
GRASS has also been used to define landslide susceptibility areas by [8]. They used a combination of GRASS with the gawk programming language to create landslide susceptibility maps of Parma River basin in Italy. They showed that very large datasets can be processed in GRASS quickly without problems.
The characterization of landscape units which are not only used in geomorpholgy but also in other scientific fields such as soil science and environmental modeling has benefited tremendously from GRASS in the past.

Geostatistics

[5] used a combination of GRASS, R and PostgreSQL to analyze various geodatasets. They showed that these techniques provide a powerful toolbox to analyze natural phenomena as well as socio-economic data.

Hydrologic Modeling

Hydrologic models like the USDA-Water Erosion Prediction Project (WEPP) model can be easily parameterized with GRASS as shown by [30]. [9] calculated a more appropriate flow time as an input for the flow analysis of a river in East Germany based on WaSiM-ETH. Besides the existing models incorporated in GRASS, custom-written models can be created as shown by [12]. They incorporated a Soil Moisture Routing model which combines elevation, soil and landuse data and predicts soil moisture, evapotranspiration, saturation-excess overland flow and interflow for a watershed.

Oceanography

For nautical hydrographic surveys GRASS offers some helpful modules to generate bathymetric surfaces by the interpolation of sounding data. [19] built up an environmental GIS database for the White Sea based on GRASS incorporating several hydrological and chemical parameters to validate numerical ecosystem modeling with the purpose to evaluate effects of climate change and human impact on this ecosystem.

Landscape Epidemiology and Public Health

With the help of GIS the spread of epidemics can be analysed or predicted. The outbreak of avian influenza in northern Italy in winter 1999–2000 was examined by [23]. GRASS and R were used to map the distribution of the outbreaks of highly pathogenic avian influenza which was caused by a H7N1 subtype virus.
To predict the risk of Lyme Disease for the Italian province of Trento GRASS has been used in several studies. The distribution of ticks infected with *Borrelia burgdorferi* s.l. was analysed by [28] with a bootstrap aggregation model of tree based classifiers in GRASS. The occurrence of ticks were cross-correlated with environmental data in GIS. [14] developed a spatial model of the probability of tick presence using machine learning techniques incorporated in GRASS and R.
A combination of GRASS, Mapserver and R is used by the Public health Applications in Remote Sensing (PHAiRS) NASA REASoN project. The objective of this project is to offer federal state and local government agencies dynamic information that might impact the spread of illnesses. Environmental and atmospheric conditions which affect public health are derived from NASA data sets and presented in a way that local public health officials can use them for decision making.

Remote Sensing

GRASS with its sophisticated raster processing capability and the already implemented image processing modules offer the user sophisticated tools for processing remote sensing data at low cost. The existing modules include functions for image preparation, image classification and image ratios. The software has also some functions for creating orthophotos and image enhancement. [26] give an overview of the tools for image processing in GRASS.
The potential to detect objects from airborne Laser Scanning data for urban mapping is described in [18].

Soil Science

GRASS is used in this field for several tasks because several useful tools have been developed by soil scientists.
Terrain parameters are important input to soil modeling and are widely used to assist in the mapping of soil properties. The aspect angle is commonly used by soil scientists as a proxy for the variation in surface moisture. Together with climatic data it is possible to derive a quantitative model of the surface soil moisture status of a landscape. The components of the solar radiation budget for each cell can be computed by GRASS using modules where site specific solar radiation is considered. [29] improved the predictive potential of pedotransfer functions. These are the basis of some hydrologic models with which soil hydraulic behavior can be characterized on a large scale. These terrain parameters included topographic information that was processed with the help of GRASS.
[1] derived a three dimensional continuous soil model with the help of GRASS. He used fuzzy sets to represent soil-landscape relations as fuzzy rules. With these rules he examined landscape information data which led to a three dimensional soil model.

Education

The GRASS community actively promotes the teaching of GRASS and other FOSSGIS (Free and Open Source Software GIS) programs to train the next generation in these forward looking techniques. Educational materials are available on the GRASS wiki.

Future Directions

The ongoing development of GRASS as a native Windows application and the implementation of a new unified Graphical User Interface for Linux, Mac, Windows and Unix using WxWidgets and Python will certainly increase distribution and use of the program. The prototype code is already written and is being tested. Its advantages in modeling, price and adaptability makes GRASS a strong alternative to other GIS software. Increasing popularity will logically lead to increased development efforts for the software. More people will contribute to the source code, bugtracking and documentation. GRASS has already some innovative functions implemented (e. g. functions for network analysis like shortest path, route planing), waiting for new applications to be developed on top. For 3D modeling the infrastructure and modules are in place for raster, vector and site data leading to an increasing usage in spatial modeling.

Cross References

- ▶ Open Geospatial Consortium
- ▶ PostGIS
- ▶ Quantum GIS
- ▶ University of Minnesota (UMN) Map Server

Recommended Reading

1. Ameskamp, M.: Three-dimensional Rule-based Continuous Soil Modelling. PhD thesis, Christian-Albrechts-Universitaet Kiel (1997)
2. Baker, W.L., Weisberg, P.J.: Using GIS to model tree population parameters in the rocky mountain national park forest-tundra ecotone. J. Biogeogr. **24**, 513–526 (1997)
3. Bivand, R.S.: Integrating grass 5.0 and r: GIS and modern statistics for data analysis. 7th Scandinavian Research Conference on Geographical Information Science. Aalborg, Denmark (1999)
4. Bivand, R.S.: Using the r statistical data analysis language on grass 5.0 GIS database files. Comp. & Geosci. **26**, 1043–1052 (2000)
5. Bivand, R.S., Neteler, M.: Open source geocomputation: using the r data analysis language integrated with grass GIS and postgresql data base systems (2000)
6. Brandt, R., Groenewoudt, B.J., Kvamme, K.L.: An experiment in archaeological site location: Modeling in the netherlands using GIS techniques. World Arch. **24**(2), 268–282 (1992)
7. Ciolli,M. Zatelli, P.: http://www.ing.unitn.it/~grass/conferences/grass2002/proceedings/proceedings/abstracts.html. online, 12 (2006)
8. Clerici, A., Perego, S., Tellini, C., Vescovi, P.: A procedure for landslide susceptibility zonation by the conditional analysis method. Geomorphol. **48**(4), 349–364 (2002)
9. Cullmann, J., Mishra, V., Peters, R.: Flow analysis with wasim-eth model parameter sensitivity at different scales. Adv. Geosci. **9**, 73–77 (2006)
10. Deuchler, C., Wählisch, M., Gehrke, S., Hauber, E., Oberst, J., Jaumann, R.: Combining mars data in grass GIS for geological mapping. ISPRS Congress, 12–23 July 2004 Istanbul, Turkey, vol.#160;35, ISPRS, Istanbul (2004)
11. Ducke, B.: Archaeological predictive modelling in intelligent network structures. Doerr, M., Sarris, A. (eds.): CAA 2002 The Digital Heritage of Archaeology. Proceedings of the 30th CAA conference held at Heraklion, Crete, Greece, 2–6 April 2002, pages 267–273, Athens (2003)
12. Frankenberger, J.R., Brooks, E.S., Walter, M.T., Walter, M.F., Steenhuis, T.S.: A GIS-based variable source area hydrology model. Hydrol. Process **13**, 805–822 (1999)
13. Frigeri, A., Federico, C., Pauselli, C., Minelli, G.: Identifying wrinkle ridges from mars mgs and viking mission data: Using grass GIS in planetary geology. Trans. GIS **8**(2), 255–262 (2004)
14. Furlanello, C., Neteler, M., Merler, S., Menegon, S., Fontanari, S., Donini, A., Rizzoli, A., Chemini, C.: GIS and the Random Forest Predictor: Integration in R for Tick-borne Disease Risk Assessment. In: Hornik, K., Leisch, F. (eds.) Proceedings of the 3rd International Workshop on Distributed Statistical Computing, March 20–22, Vienna, Austria (2003)
15. Garzon, M.B., Blazek, R., Neteler, M., Dios, R.S., Ollero, H.S., Furlanello, C.: Predicting habitat suitability with machine learning models: The potential area of pinus sylvestris l. in the iberian peninsula. Ecol. Modell. **197**(3–4), 383–393 (2006)
16. GRASS Development Team. http://grass.itc.it/. online, 12 (2006)
17. Grohmann, C.H.: Morphometric analysis in geographic infromation systems: applications of free software grass and r. Comput. & Geosci. **30**, 1055–1067 (2004)
18. Höfle, B., Rutzinger, M., Geist, T., Stötter, J.: Using airborne laser scanning data in urban data management – set up of a flexible information system with open source components. In: Proceedings UDMS 2006: Urban Data Management Symposium, Aalborg, Denmark (2006)
19. Kaitala, S., Shavykin, A., Volkov, V.A.: Environmental GIS database for the white sea. In: Proceedings of the Open source GIS – GRASS users conference, 11–13 September 2002, Trento, Italy (2002)
20. Kajiyama, A., Ikawa, N., Masumoto, S., Shiono, K., Raghavan, V.: Three-dimensional geological modeling by foss grass GIS – using some field survey data –. In: Proceedings of the FOSS/GRASS Users Conference – Bangkok, 12–14 September 2004, Bangkog, Thailand (2004)
21. Lake, M.W.: The use of pedestrian modelling in archaeology, with an example from the study of cultural learning. Environ. Plann. B: Plann. Des. **28**, 385–403 (2001)
22. Lake, M.W., Woodman, P.E., Mithen, S.J.: Tailoring GIS software for archaeological applications: An example concerning viewshed analysis. J. Arch. Sci. **25**(1), 27–38 (1998)
23. Mannelli, A., Ferre, N., Marangon, S.: Analysis of the 1999 2000 highly pathogenic avian influenza (h7n1) epidemic in the main poultry-production area in northern Italy. Prev. Vet. Med., **73**, 273–285 (2006)
24. Merlo, S., Colin, A.: Developing a multidimensional GIS framework for archaeological excavations. In: CIPA 2005, XX International Symposium Torino, Torino (2005)
25. Mitasova, H., Mitas, L., Brown, W.M., Gerdes, D.P., Kosinovsky, I., Baker, T.: Modelling spatially and temporally distributed phenomena: new methods and tools for grass GIS. Int. J. Geogr. Inf. Syst. **9**(4), 433–446 (1995)
26. Neteler, M., Grasso, D., Michelazzi, I., Miori, L., Merler, S., Furlanello, C.: An integrated toolbox for image registration, fusion and classification. Int. J. Geoinf. Special Issue on FOSS/GRASS 2004 & GIS-IDEAS 2004, **1**(1), 51–60 (2005)
27. Neteler, M., Mitasova, H.: Open Source GIS: A GRASS GIS Approach. Number 773 in SECS. Kluwer Academic Publishers/Springer, Boston, first edition, June 2001. ISBN: 1–4020-8064-6; also published as eBook: ISBN 1-4020-8065-4
28. Rizzoli, A., Merler, S., Furlanello, C., Genchi, C.: Geographical information system and bootstrap aggregation (bagging) of tree-based classifiers for lyme disease risk prediction in trentino, italian alps. J. Med. Entomol. **39**(3), 485–492 (2002)
29. Romano, N., Palladino, M.: Prediction of soil water retention using soil physical data and terrain attributes. J. Hydro. **265**, 56–75 (2002)
30. Savabi, M.R., Flanagan, D.C., Hebel, B., Engel, B.A.: Application of WEPP and GIS-grass to a small watershed in indiana. J. Soil Water Conserv. **50**(5), 477–483 (1995)
31. Tucker, K., Rushton, S.P., Sanderson, R.A., Martin, E.B., Blaiklock, J.: Modelling bird distributions – a combined GIS and bayesian rule-based approach. Landsc. Ecol. **12**(2), 77–93 (1997)

GRASS-GIS

► GRASS

Grid

Ilya Zaslavsky
San Diego Supercomputer Center, University of California San Diego, San Diego, CA, USA

Synonyms

Computational grid; Data grid; Computational infrastructure

Definition

Grid computing is a model for organizing large-scale computation and data management such that process execution tasks are distributed across many network computers, and data transmission occurs over high-speed parallel network connections, in a system that resembles parallel computer architecture. Grids rely on "resource virtualization", that is, organize different locally-managed processing and data nodes as relatively homogeneous "grid resources" that follow open standards and enable quality of service through resource sharing and service-level agreements.

Main Text

Grid computing is defined via a series of specifications being developed within the Open Grid Forum (OGF), a collaboration between industry and academia. Globus Toolkit is a middleware component implementing OGF standards. Initial ideas of grids, and grid service specifications, have been explored in seminal articles by Ian Foster, Carl Kesselman and Steve Tuecke. In recent years, grid services and open grid services architecture (OGSA) have evolved towards better compatibility with web services and service-oriented architecture principles. The Web Service Resource Framework (WSRF) is a set of web service specifications describing OGSA implementation using web services.

Several research projects have applied grid computing principles to spatial information processing. In particular, they include development of stateful grid services for assembling thematic maps within the Geosciences Network (GEON) project (support for stateful services environment is a distinct feature of grid services compared with web services). Another direction is relegating spatial information processing to P2P networks, in particular, development of decentralized indexing and routing techniques for spatial data. The goal is to enable dynamic insertion or deletion of spatial information so that processing is adjusted to the set of currently available nodes in a P2P network. Grids can primarily focus on aggregating processing power of distributed computers (computational grids), or on secure management and sharing of large data volumes (data grids), or on remote control of scientific instruments (equipment grids). However, many grid implementations share features of all the three types.

Cross References

► Cyberinfrastructure for Spatial Data Integration
► Grid
► Spatial Information Mediation
► Web Services

Grid Computing

► Grid, Geospatial

Grid, Geospatial

Aijun Chen, Liping Di, Yaxing Wei, Yuqi Bai
Center for Spatial Information Science and Systems, George Mason University, Fairfax, VA, USA

Synonyms

Geospatial data grid; Geospatial computing grid; Geospatial computational grid; Grid computing; Distributed computing

Definition

Geospatial grid refers to the *grid*, an emerging cyberinfrastructure, dealing with geospatial information (aka spatial information grid). The geospatial grid includes geospatial data grids (GDGs) and geospatial computational grids (GCGs) [1,2]. A geospatial grid leverages geospatial standards and interfaces to handle geospatial data, information, and processes. A GDG is used for registering, managing, retrieving, and securely copying and moving geospatial data and its replicas among data centers distributed around the world. A GCG focuses on processing and analyzing geospatial data and their replicas by utilizing distributed geospatial resources in the grid environment. Both GDG and GCG are based on the fundamental grid standards, the Open Grid Services Architecture (OGSA) and the Web Service Resource Framework (WSRF) [3]. A geospatial grid is an advanced, open, service-oriented, and distributed geospatial data and computing system. Figure 1 in the later of the paper illustrates an integrated system of a geospatial data grid and a geospatial computing grid. Geospatial grid services can be combined to form a service chain, which acts as a new grid service, for providing new geospatial functions. *Geospatial semantic grid* and *geospatial knowledge grid* are used to refer to the application of ontology and artificial intelligence to geospatial grids.

Historical Background

The grid is used for the next generation data- and computing-intensive information infrastructure providing a number of new capabilities that emerged in the late 1990s [4]. The Globus Toolkit, including (a) security, (b) data management, (c) information services, (d) execution management, is a de facto standard middleware for implementing grids. The grid was applied to many disciplines and research areas, such as the the Earth System Grid (ESG) project for Earth science [5], the Grid Physics Network and the Particle Physics Data Grid projects for physics [6,7]; Austria's BioGrid, the United States' Bioportal, Europe's Bioinformatics Grid, and South Africa's National Bioinformatics Grid projects for bioinformatics; and the National Virtual Observatory (NVO) for astronomy [8]. The application of the grid in geospatial field originates the term geospatial grid.

In 2001, the Chinese Ministry Of Science and Technology (MOST) sponsored a spatial information grid. The San Diego Super Computing Center developed the Spatial Resource Broker for constructing a geospatial data grid. In the early 2000s, the National Science Foundation (NSF) supported several geospatial-related information technology researches, such as the Geosience Network (GEON) (www.geongrid.org/) and Network for Earthquake Engineering Simulation (NEES) (www.nees.org/). Di [1,2] introduced the concept of a geospatial grid in 2004 through the project on "The Integration of Grid and OGC (Open Geospatial Consortium) (IGO) Technologies for Earth Science Modeling and Applications" sponsored by the National Aeronautics and Space Administration (NASA) Advanced Information System Technology (AIST) Program. Following these activities, the geospatial grid concept was used in many journal and conference papers [2,10].

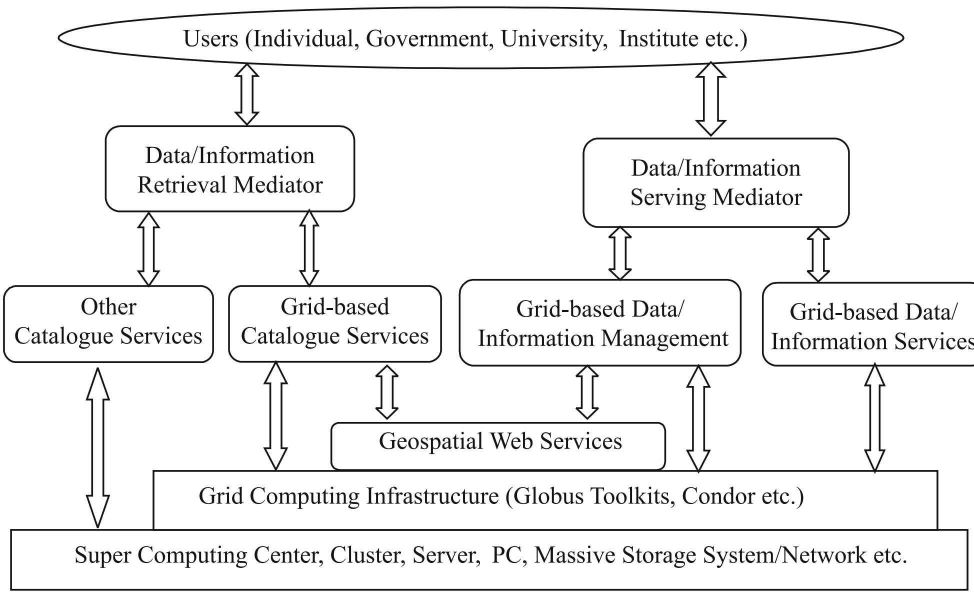
The IGO project addressed how to effectively and efficiently share, process, visualize, and serve geospatial resources using grid technology and OGC web services standards. In the project, a number of OGC web services were implemented as grid services and deployed in the Globus-based grid environment. Those OGC services included the web mapping service (WMS), the web coverage service (WCS), the catalogue service/web profile (CS/W), the web image classification service (WICS), and the web coordinate transform service (WCTS). They formed the foundation of the geospatial grid for providing basic geospatial functions such as registration, updating and retrieval of geospatial data and services, and subsetting, reprojection, reformatting, resampling, and visualization of distributed geospatial data. In addition to fully integrating grid services from the Globus Toolkit into the geospatial grid, a number of new generic grid services have also been devised and implemented to strengthen the geospatial grid.

Scientific Fundamentals

The geospatial grid provides a promising mechanism to share worldwide huge volumes of heterogeneous geospatial computing and data resources to facilitate important research programs, such as climate change, weather forecasting, and natural hazard prediction. To take full advantages of grid technology for effectively sharing distributed geospatial resources and facilitating their interoperability, a geospatial grid must be established following geospatial and Grid technology standards.

A geospatial data grid is for managing and sharing large volumes of geospatial data. The top level of a GDG architecture consists mainly of such grid services as a secure data transfer service (GridFTP), a data replica service (DRS), a replica location service (RLS), and a reliable file transfer service (RFT). A geospatial computational grid (GCG) is focused on the sharing and utilization of geospatial computing capabilities. It provides distributed supercomputing, high-throughput, on-demand, data-intensive and collaborative computing. Most of the geospatial-related applications demonstrate a tremendous appetite for computational resources (CPU, memory, storage, etc.) that cannot be met in a timely fashion by expected growth in single-system performance. OGC web service standards play a very important role in a GCG. (Fig. 1).

The architecture includes a geospatial data grid and a geospatial computational grid. Grid computing infrastructure consists of the popular grid middleware: Globus Toolkit and Condor. Geospatial management and serving grid services can be developed using geospatial web services and other geospatial services, e. g., Geographic Resources Analysis Support System (GRASS), and following both grid standards and geospatial standards. A geospatial grid catalogue service can be developed following the geospatial catalogue standard for registering and querying geospatial grid services in the grid environment. A grid retrieval mediator can be designed to coordinate the user's query for multiple catalogue services and data replica index services. All geospatial



Grid, Geospatial, Figure 1 The architecture of a geospatial grid

data/information and grid services are registered to catalog services. Geospatial data/information serving mediators are used to coordinate users' requests among all grid services and send each request to the grid services with best performance. Optimized grid service can be a pure grid service and implemented based on grid software. All components in the geospatial grid are grid service-oriented. A service to manage job submission [e. g., Web Service Globus Resource Allocation Manager (WS GRAM)] and schedule grid services (e. g., Condor) can be used to manage all submitted jobs.

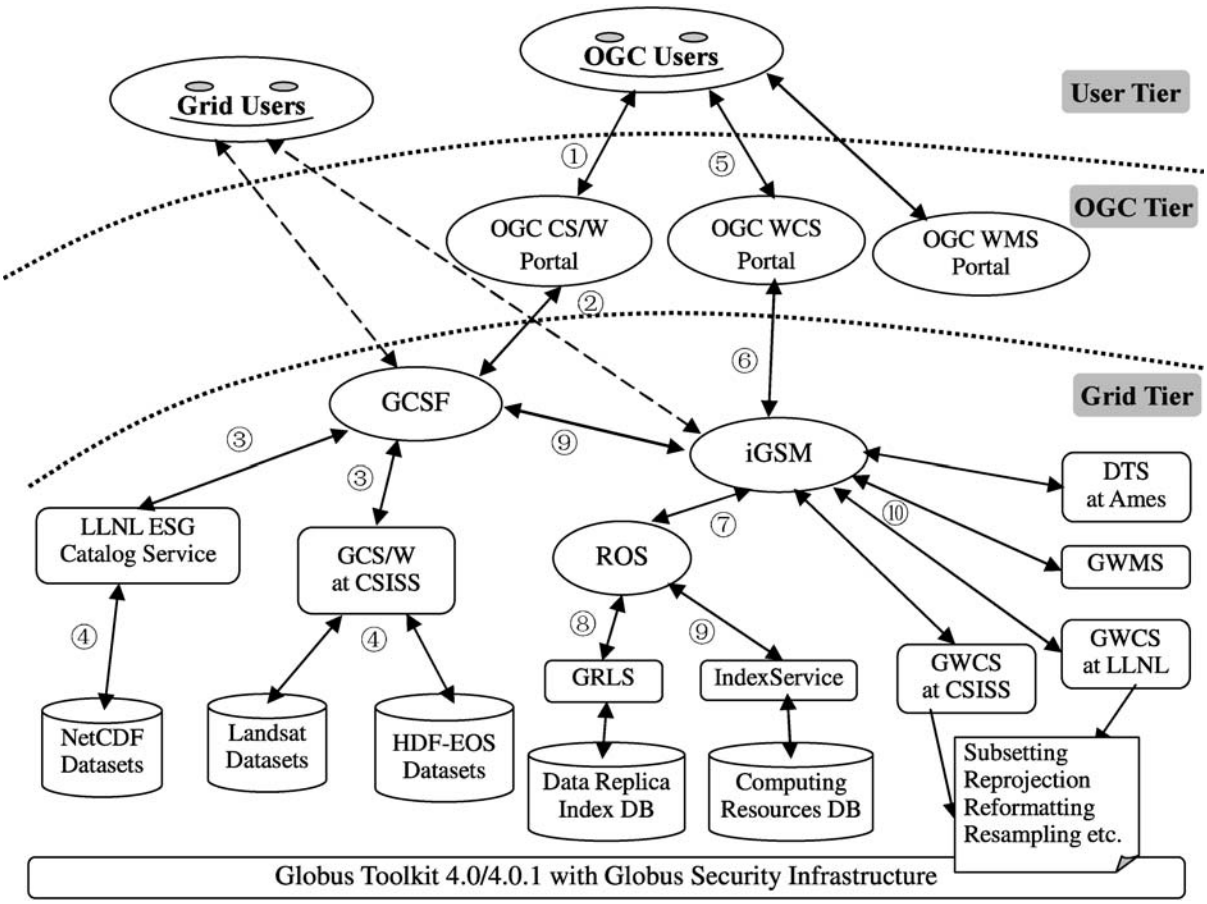
Prototype

OGC has developed a set of widely accepted spatial web services for geospatial interoperability, such as the WMS, WCS, and CS/W specifications. Currently, numerous services complying with those specifications have been developed and deployed worldwide [10]. By adopting OGC specifications and providing OGC services in the grid environment, the geospatial grid not only extends the grid capabilities to geospatial disciplines and enrich the grid computing technology, but it also makes it possible to easily use the newest computer technology available and to resolve some large geospatial questions that are difficult to resolve without the geospatial grid.

Figure 2 is the architecture and dataflow diagram of a standards-compliant geospatial grid system prototype. The prototype consists of three tiers: the user tier, the OGC tier, and the grid tier [11]. The OGC tier provides OGC-compliant clients with standard OGC interfaces that are supported by grid technology. To an OGC client, this tier acts as a set of OGC compliant servers, but to the grid tier, the OGC tier operates as an authorized user of the grid tier. Any user request compliant with OGC standards can be submitted through these interfaces and processed through grid computing resources. The grid tier provides grid users with standard grid service interfaces. Any authorized grid user in a virtual organization (VO) can access any grid resources in the same VO or any other authorized VO. Therefore, the geospatial grid hides the complexity of the grid and the managed resources. It provides standards-compliant interfaces for geospatial users and gives them transparent access to grid-enabled distributed resources.

The OGC Standard-compliant web portals, including a CS/W portal, a WCS portal and a WMS portal, are based on grid technology. Through those portals, any OGC-compliant clients can use all the resources managed by the grid.

Grid enabled GSW (GCS/W) [12] is based on the OGC catalog service specification. The information model of OGC CS/W has been extended to satisfy the requirement for describing Earth observation data while complying with the International Organization for Standardization (ISO), the Federal Geographic Data Committee (FGDC), and NASA EOS (Earth Observation System) Core System (ECS) metadata standards [13]. The Grid-Enabled Catalog Service Federation (GCSF) acts as the catalogue mediator. It aims to federate different catalog services into one logical service that provides a standard uniform interface to users. The GCSF is designed to integrate all grid-enabled



Grid, Geospatial, Figure 2 Architecture and dataflow of an open standard compliant geospatial grid. *CS/W* Catalog service for web, *WCS* web coverage service, *GWCS* grind-enabled WCS *WMS* web map service, *GWMS* grid-enabled WMS, *GCSF*: grid-enabled Catalog Service Federation, *iGSM* intelligent grid service mediator, *ROS* replica and optimization service, *DTS* data transfer service, *GRLS* grid replica location service, *ESG* Earth System Grid, *LLNL* Lawrence Livermore National Laboratory, *Ames* NASA Ames Research Center, *CSISS* Center for Spatial Information Science and Systems

catalog services within a Virtual Organization (VO). In the prototype system, the GCSF has federated the ESG catalog services from LLNL (Lawrence Livermore National Laboratory), GCS/W from CSISS at GMU, and NASA ECHO (EOS ClearingHOuse). The GCSF mainly maps metadata, transfers protocol-based XML documents, and merges results.

The grid-enabled web coverage service (GWCS) can be accessed indirectly by any OGC user through the WCS portal and directly by authorized grid users. The main customizable data access interface of GWCS is getCoverage. GWCS contains some built-in geoprocessing functions, such as subsetting, reprojection, reformatting, and georectification. These functions provide flexible operations with which the users can customize their desired data. The grid-enabled web mapping service (GWMS), on users' requests, dynamically produces static maps from spatially referenced geospatial data managed by the grid. The getMap operation is grid-enabled to produce maps in the formats of PNG (Portable Network Graphics), GIF (Graphics Interchange Format), or JPEG (Join Photographic Experts Group).

With the grid computing infrastructure, all OGC standard interfaces are implemented in the prototype system. By collaborating with the intelligent grid service mediator (iGSM), the replica and optimized service (ROS) and the data transfer service (DTS), accelerated processing speed, better serving performance, and data quality are obtained.

The intelligent grid service mediator (iGSM) is a grid service. It receives and responds to user requests from the OGC WCS/WMS portal, and mediates the received user requests among ROS, DTS, GCS/W, GWCS, and GWMS. The iGSM service sends the user request, including the logical data address, to ROS to get back the best performing service that serves real data. If no service is returned by ROS, iGSM will query GCS/W to get all the GWCS/GWMS services in the VO and all replicas of the required data and then select the best service to serve the data with the best performance. When data transfer between two machines in the VO is required, DTS is invoked with the authentication operation.

ROS is used to optimize user requests for data and services based on the grid technology. It provides a mechanism for managing grid-enabled geospatial data replicas and the best service and data selection mechanism. It integrates the Globus Toolkit's IndexService and Replica Location Service (RLS). The IndexService is used to obtain dynamic information for every machine that hosts GWCS/GWMS. Meanwhile, RLS manages all distributed replicas of every registered geospatial data item. ROS will identify all machines that can serve the data the user requires and then select the one most appropriate for their current resources available, e. g., CPU, memory, storage.

DTS works as a grid service to be invoked by other grid services when secure data transfer between different machines in the same VO or a different VO is needed. This data transfer is secured strictly through the grid security infrastructure (GSI). Both a user certificate and a host certificate are involved in identifying the authorization and authentication. It is based on the GridFTP, which is a command line component of the Globus Toolkit. Grid-enabled RLS (GridRLS), is based on the Globus RLS and is a grid service that enables RLS to orchestrate with any other grid services. RLS is a component of the Globus Toolkit. It can be run from the command line, but it cannot be run as a grid service in a Globus Toolkit 3.0 or above environment.

Key Applications

The geospatial grid can be applied in many geospatial science domains for large-scale data- and computing-intensive research and applications to acquire data, deploy services, construct models, execute workflows, and disseminate results. Some applications are global climate change, weather forecasting, crop yield, natural hazard, and air quality, etc.

Global Climate Change

Global climate change is a change in the long-term weather patterns that characterize the regions of the world. Research on global climate change requires analysis of large volumes of historical and current data from sensors aboard various platforms in addition to field observations. Computing-intensive models are routinely used in such research. The required massive quantities of geospatial data and large computing capabilities can be made available through the geospatial grid.

Geospatial Product Virtualization

In the geospatial grid, users can register their own geospatial grid services in the system catalog. They can use existing grid services to construct service chain to express geoprocessing workflows and register the workflows into the catalog as virtual geospatial products [14]. The virtual geospatial products can be materialized by executing the workflows in the grid-enabled workflow engine when users request them. The workflow associated with a virtual geospatial product can be considered to be a new grid service that can be chained into another workflow. For example, in calculating landslide susceptibility, a user can create Slope and Aspect grid services and register them into the catalogue. Then, the user can compose a landslide susceptibility workflow based on the Slope service, Aspect service, GCS/W, GWCS, and GWCTS. The composition of the workflow is based on users' knowledge: GCS/W can help users to find the digital elevation model (DEM) data they need, and GWCS can serve this DEM data to the Slope and Aspect services. The intermediate results produced can be further processed through the landslide susceptibility service, and finally, the landslide susceptibility parameters can be obtained for landslide forecasting [14].

Future Directions

The advancement of information technology has meant the accumulation of increasing amount of digitized data/information/knowledge (DIK), of which 80% are geospatially related. Currently both DIK and computing resources are not fully utilized. It is expected that research facilitating the application of grid technology to derive DIK, especially geospatial, will become increasingly important.

The grid-enabled collaborative and intelligent environments for geospatial computing and modeling that can handle huge amount of shared geospatial resources will be a focus of future research. Ontology-driven semantic workflow is a key technology for modeling geospatial applications in the Geospatial Grid.

The Geospatial Grid not only offers conventional functions of geographical information systems, it also provides an open, ubiquitous and intelligent platform for authorized users to share their knowledge and innovative ideas with the community and quickly convert the knowledge and ideas into operational capabilities within the Geospatial Grid.

Recommended Reading

1. Di, L.: Geospatial grids, geospatial modeling, and virtual geospatial products. In: Earth Observation and Geospatial Technology

G

for Civil Society, University College London, London, 23–25 June 2004
2. Di, L.: The geospatial grid. In: S. Rana, J. Sharma (eds.) Frontiers of Geographic Information Technology, pp 121–138. Springer-Verlag, Berlin, Heidelberg (2006)
3. Globus (2006) Globus Toolkit 4.0 Release Manuals. http://www.globus.org/toolkit/docs/4.0/. Accessed 22 Sept 2006
4. Chen, A., Di, L., Bai, Y., Wei, Y.: Grid computing and its application to geoinformatics. In: Karimi, H. (ed.) The Encyclopedia of Geoinformatics. Idea Group Inc., Hershey, PA, USA (2006, in press)
5. ESG (2006) Earth System Grid. https://www.earthsystemgrid.org/home/publicHomePage.do. Accessed 10 Sept 2006
6. GriPhyN (2006) Grid Physics Network. http://www.griphyn.org/projinfo/intro/appl_domains.php. Accessed 25 Sept 2006
7. PPDG (2006) Particle Physics Data Grid. http://www.ppdg.net/. Accessed 26 Sept 2006
8. NVO (2006) National Virtual Observatory. http://www.us-vo.org/grid.cfm. Accessed 28 Sept 2006
9. Di, L., Yang, W., Chen, A., Liu, Y., Wei, Y., Mehrotra, P., Hu, C., Williams, D.: The development of a geospatial grid by integrating OGC Web Services with Globus-based grid technology. In: GridWorld/GGF15, Boston, MA, October 3–6, 2005
10. Chen, A., Di, L., Bai, Y., Wei, Y.: An optimized grid-based, OGC standards-compliant collaborative software system for serving NASA geospatial data. In: The 2nd IEEE Systems and Software Week (SASW 2006), 30th Annual Software Engineering Workshop. Loyola College Graduate Center, Columbia, MD, USA. April 25–27, 2006
11. Di, L., Chen, A., Yang, W., Liu, Y., Wei, Y., Mehrotra, P., Hu, C., Williams, D.: The development of a geospatial data grid by integrating OGC Web Services with Globus-based grid technology. Concurrency and Computation: Practice and Experience (2006)
12. Chen, A., Di, L., Wei, Y., Liu, Y., Bai, Y., Hu, C., Mehrotra, P.: Grid enabled geospatial catalogue web services. In: American Society for Photogrammetry and Remote Sensing 2005, Baltimore, MD, March 7–11, 2005
13. Wei, Y., Di, L., Zhao, B., Liao, G., Chen, A.: Transformation of HDF-EOS metadata from the ECS model to ISO 19115-based XML. Comput. Geosci. **33**(2):238–247 (2007)
14. Di, L., Chen, A., Bai, Y., Wei, Y.: Implementation of geospatial product virtualization in grid environment. In: The Sixth Annual Earth Science Technology Conference – ESTC2006, College Park, MD, June 27–29, 2006

Gridded Data

► Oracle Spatial, Raster Data

G-Ring

► Spatial Data Transfer Standard (SDTS)

Group Decisions

► Participatory Planning and GIS

Group Spatial Decision Support Systems

► Geocollaboration

GSDSS

► Geocollaboration

GSN

► Geosensor Networks

Hamiltonian Cycle with the Least Weight

► Traveling Salesman Problem (TSP)

Health Geographics

► Spatial Uncertainty in Medical Geography: A Geostatistical Perspective

Heterogeneity

► Cyberinfrastructure for Spatial Data Integration
► Ontology-Based Geospatial Data Integration

Hierarchical Data Structures

► Generalization, On-the-Fly

Hierarchical Dynamic Spatio-temporal Models

► Hierarchical Spatial Models

Hierarchical Spatial Models

ALI ARAB[1], MEVIN B. HOOTEN[2],
CHRISTOPHER K. WIKLE[1]
[1] Department of Statistics, University of Missouri-Columbia, Columbia, MO, USA
[2] Department of Mathematics and Statistics, Utah State University, Logan, UT, USA

Synonyms

Hierarchical dynamic spatio-temporal models; Geostatistical models; Hierarchies; Autoregressive models; Process model

Definition

A hierarchical spatial model is the product of conditional distributions for data conditioned on a spatial process and parameters, the spatial process conditioned on the parameters defining the spatial dependencies between process locations, and the parameters themselves.

Historical Background

Scientists across a wide range of disciplines have long recognized the importance of spatial dependencies in their data and the underlying process of interest. Initially due to computational limitations, they dealt with such dependencies by randomization and blocking rather than the explicit characterization of the dependencies in their models. Early developments in spatial modeling started in the 1950's and 1960's motivated by problems in mining engineering and meteorology [11], followed by the introduction of Markov random fields [2]. The application of hierarchical spatial and spatio-temporal models have become increasingly popular since the advancements of computational techniques, such as MCMC methods, in the later years of the 20th century.

Scientific Fundamentals

Methods for spatial and spatio-temporal modeling are becoming increasingly important in the environmental sciences and other sciences where data arise from processes in spatial settings. Unfortunately, the application of traditional covariance-based spatial statistical models is either inappropriate or computationally inefficient in many problems. Moreover, conventional methods are often incapable of allowing the researcher to quantify uncertainties corresponding to the model parameters since the parameter space of most complex spatial and spatio-temporal models is very large.

Hierarchical Models

A main goal in the rigorous characterization of natural phenomena is the estimation and prediction of processes as well as the parameters governing processes. Thus a flexible framework capable of accommodating complex relationships between data and process models while incorporating various sources of uncertainty is necessary. Traditional likelihood based approaches to modeling have allowed for scientifically meaningful data structures, though, in complicated situations with heavily parameterized models and limited or missing data, estimation by likelihood maximization is often problematic or infeasible. Developments in numerical approximation methods have been useful in many cases, especially for high-dimensional parameter spaces (e. g., Newton-Raphson and E-M methods, [3]), though can still be difficult or impossible to implement and have no provision for accommodating uncertainty at multiple levels.

Hierarchical models, whereby a problem is decomposed into a series of levels linked by simple rules of probability, assume a very flexible framework capable of accommodating uncertainty and potential a priori scientific knowledge while retaining many advantages of a strict likelihood approach (e. g., multiple sources of data and scientifically meaningful structure). The years after introduction of the Bayesian hierarchical model and development of MCMC (i. e., Markov Chain Monte Carlo) have brought on an explosion of research, both theoretical and applied, utilizing and (or) developing hierarchical models.

Hierarchical modeling is based on a simple fact from probability that the joint distribution of a collection of random variables can be decomposed into a series of conditional models. For example, if a, b, c are random variables, then basic probability allows the factorization:

$$[a, b, c] = [a|b, c][b|c][c] \,, \tag{1}$$

where the notation [.] is used to specify a probability distribution and $[x|y]$ refers to the distribution of x conditioned on y. In the case of spatial and spatio-temporal models, the joint distribution describes the behavior of the process at all spatial locations of potential interest (and, possibly, all times). This joint distribution (left hand side of (1) is difficult to specify for complicated processes. Typically, it is much easier to specify the distribution of the conditional models (right hand side of (1)). In that case, the product of the series of relatively simple conditional models gives a joint distribution that can be quite complex.

When modeling complicated processes in the presence of data, it is helpful to write the hierarchical model in three basic stages:

Stage 1. Data Model: [*data*|*process, data parameters*]
Stage 2. Process Model: [*process*|*process parameters*]
Stage 3. Parameter Model: [*data and process parameters*].

The basic idea is to approach the complex problem by breaking it into simpler subproblems. Although hierarchical modeling is not new to statistics [4], this basic formulation for modeling complicated spatial and spatio-temporal processes in the environmental sciences is a relatively new development (e. g., [5,6]). The first stage is concerned with the observational process or "data model", which specifies the distribution of the data given the fundamental process of interest and parameters that describe the data model. The second stage then describes the process, conditional on other process parameters. Finally, the last stage models the uncertainty in the parameters, from both the data and process stages. Note that each of these stages can have many sub-stages.

Implementation

The goal is to estimate the distribution of the process and parameters given the data. Bayesian methods are naturally suited to estimation in such hierarchical settings, although non-Bayesian methods can sometimes be utilized but often require additional assumptions. Using a Bayesian approach, the "posterior distribution" (i. e., the joint distribution of the process and parameters given the data) is obtained via Bayes' Theorem:

$$\begin{aligned} &[\textit{process, parameters}|\textit{data}] \\ &\propto [\textit{data}|\textit{process, parameters}] \\ &\qquad \times [\textit{process}|\textit{parameters}][\textit{parameters}] \,. \end{aligned} \tag{2}$$

Bayesian statistics involves drawing statistical conclusions from the posterior distribution which is proportional to the data model (i. e., the likelihood) times the a priori knowledge (i. e., the prior). Bayes' theorem is thus the mechanism that provides access to the posterior. Although simple in principle, the implementation of Bayes' theorem for complicated models can be challenging. One challenge concerns the specification of the parameterized component distributions on the right-hand side of (2). Although there has long been a debate in the statistics community concerning the appropriateness of "subjective" specification of such distributions, such choices are a natural part of scientific modeling. In fact, the use of scientific knowledge in the prior distribution allows for the incorporation of uncertainty related to these specifications explicitly in the model. Another, perhaps more important, challenge, from a practical perspective, is the calculation of the posterior distribution. The complex and high-dimensional nature of many scientific models (and indeed, most spatio-temporal models) prohibits the direct evaluation of the posterior. However, MCMC approaches can be utilized to estimate the posterior distribution through iterative sampling. As previously mentioned, the use of MCMC has been critical for the implementation of Bayesian hierarchical models, in that realistic (i. e., complicated) models can be considered; this is especially evident in the analysis of spatial and spatio-temporal processes. Yet, typically the computational burden must be considered when formulating the conditional models in such problems. Thus, the model building phase requires not only scientific understanding of the problem, but in what ways that understanding can be modified to fit into the computational framework.

Non-analytical hierarchical models can be fitted to data using high-level programming languages (such as R, S-plus, MATLAB) or low-level languages (such as C, C++, FORTRAN). High-level languages allow for efficient programming, whereas low-level languages often allow for more efficient execution. Alternatively, the freely-distributed Bayesian computation software WinBUGS (http://www.mrc-bsu.cam.ac.uk/bugs/) and its spatial package GeoBUGS can be used to carry out simple Bayesian computations [7].

Spatial Processes

In this section the focus is on the process model stage of the hierarchical framework described in the previous section and specifically applied in spatial settings. The two important cases of continuous and areal data are considered, and popular modeling choices and their hierarchical forms are discussed.

General Hierarchical Spatial Model Framework

In order to consider spatial models in a general hierarchical framework, assume $\boldsymbol{Z} = (Z(r_1), \ldots, Z(r_m))'$ as a vector of observations of a spatial process denoted by $\boldsymbol{y} = (y(s_1), \ldots, y(s_n))'$ where the spatial locations of the observations (r_i) do not necessarily correspond to the support of the underlying spatial process of interest (s_i). The general framework for hierarchical spatial modeling is given by the following simple and flexible structure based on the three-stage component models described previously:

$$[\boldsymbol{Z} \mid \boldsymbol{\mu}, \boldsymbol{\theta}_z][\boldsymbol{\mu} \mid \boldsymbol{y}, \boldsymbol{\theta}_\mu][\boldsymbol{\theta}_z, \boldsymbol{\theta}_\mu] \,,$$

where θ_z and $\boldsymbol{\theta}_\mu$ are parameter vectors. The data model typically specifies that the measurements are from an underlying "true" process in the presence of a measurement error process. A generalized linear framework for the process component can be considered which generalizes the Gaussian model to the exponential model (see [8]). Thus the process model can be written,

$$h(\boldsymbol{\mu}) = \boldsymbol{X\beta} + \boldsymbol{Uy} + \boldsymbol{\eta} \,,$$

where $h(.)$ is a known link function, $\boldsymbol{\mu}$ is the mean spatial process, $\boldsymbol{X} = (\mathbf{x}'(s_1), \ldots, \mathbf{x}'(s_n))'$ denotes the covariates, $\boldsymbol{U}$ is a known matrix for the spatial random field $\boldsymbol{y}$, and $\boldsymbol{\eta}$ is process model noise. This process specification can easily accommodate most conventional spatial models.

For example, in case of normal data, the data model can be written as,

$$\boldsymbol{Z} \mid \boldsymbol{y}, \boldsymbol{\Sigma} \sim N(\boldsymbol{Ky}, \boldsymbol{\Sigma}) \,,$$

where $\boldsymbol{Z}$ denotes measurements from an underlying "true" process $\boldsymbol{y}$ in the presence of a measurement error process, and $\boldsymbol{K}$ is a matrix that maps the observations to process locations (allowing for differing observation and process supports). Similarly for count data arising from an appropriate distribution such as the Poisson, the data model can be written as,

$$\boldsymbol{Z} \mid \boldsymbol{\lambda} \sim Poisson(\boldsymbol{K\lambda}) \,,$$

where the observations $\boldsymbol{Z} = (Z(s_1),\ldots,Z(s_n))'$ are assumed to be conditionally independent, and $\lambda = (\lambda(s_1),\ldots,\lambda(s_n))'$ is the unknown spatially-varying Poisson intensity. The Poisson intensity process can then be modeled in the process stage using covariates or latent variables (see e. g., [9]).

Process Models for Spatially Continuous Data

Spatially continuous data (also known as geostatistical data) refer to spatially-indexed data at location s, where s varies continuously over some region R. The modeling of spatially continuous data has long been the dominant theme in spatial statistics. The most common class of models for spatially continuous data are known as Kriging models which are extensions of the method of minimizing the mean squared error in spatial settings. A general Kriging model has the following form:

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\eta} \,, \quad \text{where } \boldsymbol{\eta} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma}) \,, \tag{3}$$

where $\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix}$, $\boldsymbol{y}_1 = (y_{s_1}, \ldots, y_{s_n})'$ represents the spatial process at locations for which there are data, and $\boldsymbol{y}_2$ denotes the process at a new location (e. g., s_0) or locations. Furthermore, the term $\boldsymbol{X}\beta(= \boldsymbol{\mu})$ represents the mean of the process (possibly explained by covariates $\boldsymbol{X}$), and spatially correlated error $\boldsymbol{\eta}$ with covariance,

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix} .$$

Kriging then involves finding the best linear predictor of $\boldsymbol{y}_2$ given $\boldsymbol{y}_1$:

$$\boldsymbol{y}_2 \mid \boldsymbol{y}_1 \sim N(\boldsymbol{\mu_2} + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}(\boldsymbol{y}_1 - \boldsymbol{\mu}_1), \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}),$$

where $\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}$, $\mathrm{E}(\boldsymbol{y}_1) = \boldsymbol{\mu}_1$, and $\mathrm{E}(\boldsymbol{y}_2) = \boldsymbol{\mu}_2$.
The model in (3) is known as a universal Kriging model. A hierarchical representation of this model can be written as:

$$\boldsymbol{Z} \mid \boldsymbol{y}, \sigma^2 \sim N(\boldsymbol{K}\boldsymbol{y}, \sigma^2 \boldsymbol{I}) \,, \tag{4}$$

$$\boldsymbol{y} \mid \boldsymbol{\beta}, \boldsymbol{\Sigma}_y \sim N(\boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{\Sigma}_y) \,, \tag{5}$$

$$\boldsymbol{\beta} \mid \boldsymbol{\beta}_0, \boldsymbol{\Sigma}_\beta \sim N(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_\beta) \,, \tag{6}$$

$$\{\sigma^2, \boldsymbol{\theta}_y, \boldsymbol{\beta}_0, \boldsymbol{\Sigma}_\beta\} \sim [\sigma^2, \boldsymbol{\theta}_y, \boldsymbol{\beta}_0, \boldsymbol{\Sigma}_y, \boldsymbol{\Sigma}_\beta] \,. \tag{7}$$

Process Models for Areal Data

Areal data (also known as lattice data) are spatially-indexed data associated with geographic regions or areas such as counties or zip codes, and are often presented as aggregated values over an areal unit with well-defined boundaries. Spatial association among the areal units is specified by defining neighborhood structure for the areas (regular or irregular) of interest. Examples of such data include a wide variety of problems from disease mapping in counties to modeling air-pollution on a grid. Models described in this section are based on Markov random fields (MRFs). MRFs are a special class of spatial models that are suitable for data on discrete (countable) spatial domains in which a joint distribution of y_i (for $i = 1, \ldots, n$, where y_i is the spatial process at spatial unit i) is determined by using a set of locally specified conditional distributions for each spatial unit conditioned on its neighbors. MRFs include a wide class of spatial models, such as auto-Gaussian models for spatial Gaussian processes, auto-logistic models for binary spatial random variables, auto-Gamma models for non-negative continuous processes, and auto-Poisson models for spatial count processes. Here the focus is on two popular auto-Gaussian models, CAR and SAR models.

Conditionally Autoregressive (CAR) Models

As introduced by [2], conditionally autoregressive (CAR) models are popular hierarchical spatial models for use with areal data. Here, the Gaussian case is considered. Assume $y_i \equiv y(i\text{th spatial area})$ and $\boldsymbol{y} = (y_1, \ldots, y_n)'$, then the CAR model can be defined using the following n conditional distributions:

$$y_i \mid y_j, \tau_i^2 \sim N\Big(\mu_i + \sum_{j \in N_i} c_{ij}(y_j - \mu_j),\, \tau_i^2\Big), \quad i, j = 1, \ldots, n, \tag{8}$$

where N_i is defined as a set of neighbors of area i, $E(y_i) = \mu_i$, τ_{i_2} is the conditional variance, and the c_{ij} are constants such that $c_{ii} = 0$ for $i = 1, \ldots, n$, and $c_{ij}\tau_{j_2} = c_{ji}\tau_{i_2}$. It can be shown (see [1]) that the joint distribution of $\boldsymbol{y}$ using the conditional distributions can be written as

$$\boldsymbol{y} \sim N\Big(\boldsymbol{\mu}, (\boldsymbol{I} - \boldsymbol{C})^{-1}\boldsymbol{M}\Big) \,, \tag{9}$$

where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)'$, $\boldsymbol{C} = [c_{ij}]_{n \times n}$, and $M = \mathrm{diag}(\tau_1{}^2, \ldots, \tau_\mathrm{n}{}^2)$. Note that $(\boldsymbol{I} - \boldsymbol{C})$ must be invertible and $(\boldsymbol{I} - \boldsymbol{C})^{-1}\boldsymbol{M}$ must be symmetric and positive-definite.
The implementation of the CAR model is convenient in hierarchical Bayesian settings because of the explicit conditional structure. Perhaps the most popular implementation of the CAR model is the pairwise difference formulation proposed by [10], where $\boldsymbol{C}$ is decomposed into an adjacency matrix and a diagonal matrix containing information on the number of neighbors for each of the areal units which results in a simple and easy to fit version of the model. Although the convenient specification of CAR models make them attractive for modeling areal data, the usage of these models often involves numerous theoretical and computational difficulties (e. g., singularity of the covariance function of the joint distribution results in the joint distribution being improper; for more details see [7]). Several methods to overcome such difficulties have been proposed (e. g., [1,7]), however the development of strategies to address the difficulties of CAR models is a topic of ongoing research.

Simultaneous Autoregressive (SAR) Models

Simultaneous autoregressive (SAR) models, introduced by [11], are another class of spatial models for areal data. SAR models are a subset of MRFs and are popular in econometrics. Here, the Gaussian case is considered,

$$y_i = \mu_i + \sum_j b_{ij}(y_j - \mu_j) + \varepsilon_i \,, \quad i, j = 1, \ldots, n \,, \tag{10}$$

or equivalently, in matrix notation,

$$\boldsymbol{y} = \boldsymbol{\mu} + (\boldsymbol{I} - \boldsymbol{B})^{-1}\boldsymbol{\varepsilon} \,,$$

where $\boldsymbol{B} = [b_{ij}]_{n \times n}$ is a matrix that can be interpreted as the spatial-dependence matrix, $\boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \boldsymbol{\Lambda})$, and $\boldsymbol{\Lambda}$ is an $n \times n$ diagonal covariance matrix of $\boldsymbol{\varepsilon}$ (e. g., $\boldsymbol{\Lambda} = \sigma^2\, \boldsymbol{I}$). Thus, $\boldsymbol{\varepsilon}$ induces the following distribution for $\boldsymbol{y}$,

$$\boldsymbol{y} \sim N\Big(\boldsymbol{\mu}, (\boldsymbol{I} - \boldsymbol{B})^{-1}\boldsymbol{\Lambda}(\boldsymbol{I} - \boldsymbol{B}')^{-1}\Big) \,,$$

where $(\boldsymbol{I} - \boldsymbol{B})$ must be full rank. There are two common choices for $\boldsymbol{B}$; one is based on a spatial autoregression parameter (ρ) and an adjacency matrix, and the other is based on a spatial autocorrelation parameter (α) and a normalized adjacency matrix. Thus, the following alternative models can be considered:

$$y_i = \rho \sum_{j \in N_i} w_{ij} y_j + \varepsilon_i \,,$$

or,

$$y_i = \alpha \sum_{j \in N_i} \frac{w_{ij}}{\sum_k w_{ik}} y_j + \varepsilon_i$$

where the w_{ij}s are elements of an adjacency matrix $\boldsymbol{W}$, with 0 or 1 entries, describing the neighborhood structure for each unit. CAR and SAR models are equivalent if and only if their covariance matrices are equal (i. e., $(\boldsymbol{I}-\boldsymbol{C})^{-1}\boldsymbol{M} = (\boldsymbol{I}-\boldsymbol{B})^{-1}\boldsymbol{\Lambda}(\boldsymbol{I}-\boldsymbol{B}')^{-1}$). Any SAR model can be represented as a CAR model but the converse is not necessarily true. One main difference between SAR and CAR models is that the spatial-dependence matrix for CAR models ($\boldsymbol{C}$) is symmetric, while the spatial-dependence matrix for SAR models ($\boldsymbol{B}$) need not be symmetric. Although this might be interpreted as an advantage for SAR models in situations where the spatial dependence of neighboring sites is defined in an asymmetric way, non-identifiability issues related to estimation of model parameters make CAR models more preferable for cases with symmetric dependency structures (for more details on the comparisons between SAR and CAR models see [1], pages 408–410).

Spatio-temporal Processes

Spatio-temporal processes are often complex, exhibiting different scales of spatial and temporal variability. Such processes are typically characterized by a large number of observations and prediction locations in space and time, differing spatial and temporal support, orientation and alignment (relative to the process of interest), and complicated underlying dynamics. The complexity of such processes in "real-world" situations is often intensified due to non-existence of simplifying assumptions such as Gaussianity, spatial and temporal stationarity, linearity, and space-time separability of the covariance function. Thus, a joint perspective for modeling spatio-temporal processes, although relatively easy to formulate, is challenging to implement. On the contrary, a hierarchical formulation allows the modeling of complicated spatial and temporal structures by decomposing an intricate joint spatio-temporal process into relatively simple conditional models. The main advantage of the Bayesian hierarchical model over traditional covariance-based methods is that it allows the complicated structure to be modeled at a lower level in the hierarchy, rather than attempting to model the complex joint dependencies.

General Spatio-temporal Model

Let $Z(s,t)$ be a spatio-temporal process where $s \in D_s$, and D_s is a continuous or discrete, and potentially time-varying, spatial domain and $t \in D_t$ is a discrete temporal domain. The generality of the definition of the spatial domain allows for the spatio-temporal process to be applicable to both cases of continuous data and areal data. A general decomposition of the process (where $Z(s,t)$ and $Y(s,t)$ have the same spatial support and no missing data) can be written as:

$$Z(s,t) = Y(s,t) + \varepsilon(s,t) \,, \qquad (11)$$

where $Y(s,t)$ is the "true" underlying correlated process of interest and $\varepsilon(s,t)$ is a zero-mean measurement error process. The underlying process $Y(s,t)$ can be further decomposed into a mean process, additive error process, and spatial or temporal random effects. Recent approaches to spatio-temporal modeling have focused on the specification of joint space-time covariance structures (e. g., [12,13]). However, in high-dimensional settings with complicated non-linear spatio-temporal behavior, such covariance structures are very difficult to formulate. An alternative approach to modeling such complicated processes is to use spatio-temporal dynamic models in a hierarchical fashion.

Spatio-temporal Dynamic Models

Many spatio-temporal processes are dynamic in the sense that the current state of the process is a function of the previous states. There are many examples of spatio-temporal models with dynamic components in the literature (e. g., [6,9,14,15]). The joint spatio-temporal process can be factored into conditional models based on a Markovian assumption:

$$[\mathbf{Y} \mid \boldsymbol{\theta}_t, t = 1, \ldots, T] = [\mathbf{y}_0] \prod_{t=1}^{T} [\mathbf{y}_t | \mathbf{y}_{t-1}, \boldsymbol{\theta}_t] \,, \qquad (12)$$

where $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots \boldsymbol{y}_T)$, $\mathbf{y}_t = (y(s_1,t), \ldots, y(s_n,t))'$ and the conditional distribution $[\mathbf{y}_t | \mathbf{y}_{t\text{-}1}, \boldsymbol{\theta}_t]$ depends on a vector of parameters $\boldsymbol{\theta}_t$ which govern the dynamics of the spatio-temporal process of interest. An example of such spatio-temporal dynamic models is when the process has a first-order Markovian structure:

$$\mathbf{y}_t = \mathbf{H}_{\theta_t} \mathbf{y}_{t-1} + \boldsymbol{\eta}_t \,, \text{ where } \boldsymbol{\eta}_t \sim N(\mathbf{0}, \boldsymbol{\Sigma}_\eta) \,, \qquad (13)$$

where $\boldsymbol{\eta}_t$ is a spatial error process, and H_{θ_t} is a "propagator matrix" (sometimes called a transition or evolution matrix) which includes the parameters that govern the dynamics of the process. If these parameters are known or easy to estimate, an implementation of the model through Kalman filtering is possible (e. g., [15]). If the parameters are unknown, $\boldsymbol{H}_{\theta_t}$ can be modeled in a hierarchical fashion by specifying prior distributions for $\boldsymbol{H}_{\theta_t}$ or its parameters $\boldsymbol{\theta}_t$. The hierarchical form for spatio-temporal dynamic models is sometimes motivated by partial differential equations (PDEs) that describe an approximate behavior of underlying physical processes (e. g., [9]).

Key Applications

Technological advances in remote sensing, monitoring networks, and other methods of collecting spatial data in recent decades have revolutionized scientific endeavor in fields such as agriculture, climatology, ecology, economics, transportation, epidemiology and health management, as well as many other areas. However, such technological advancements require a parallel effort in the development of techniques that enable researchers to make rigorous statistical inference given the wealth of new information at hand. The advancements of computational techniques for hierarchical spatial modeling in the last two decades has provided a flexible modeling framework for researchers to take advantage of available massive datasets for modeling complex problems.

Future Directions

In this article, a brief overview of hierarchical spatial and spatio-temporal models is presented. In the recent decades, hierarchical models have drawn the attention of scientists in many fields and are especially suited to studying spatial processes. Recent computational advances and the development of efficient algorithms has provided the tools necessary for performing the extensive computations involved in hierarchical modeling. Advances in hierarchical modeling have created opportunities for scientists to take advantage of massive spatially-referenced databases. Although the literature on hierarchal spatial modeling is rich, there are still many problems and issues yet to be considered. Below, some of these challenges are briefly reviewed.

In most spatial and spatio-temporal processes, researchers have to deal with data obtained by different sources and as well as different scales. For example, a combination of Eulerian and Lagrangian data is often collected in sciences such as oceanography. Alignment and change of spatial support often presents a significant challenge for analysts. There is a need for the development of efficient methods to address these issues.

Spatial and spatio-temporal models have recently been extended to accommodate multivariate situations (e. g., popular univariate models such as Kriging and CAR models have been extended to include multivariate cases). The distinction between continuous data and areal data, as described for the univariate case, holds true for the multivariate case. Multivariate approaches have the added advantage of not only being able to rely on covariate and covariance based information, but to "borrow strength" between observation vectors (i. e., response variables) as well. Examples of such multivariate models are cokriging for multivariate continuous data and multivariate CAR models for areal data, multivariate dynamic models.

Spatial and spatio-temporal models are typically high-dimensional. This characteristic complicates the modeling process and necessitates development of efficient computational algorithms on one hand, and implementation of dimension reduction methods (e. g., recasting the problem in a spectral context), on the other hand.

Cross References

► Constraint Databases and Data Interpolation
► Hurricane Wind Fields, Multivariate Modeling
► Spatial and Geographically Weighted Regression
► Statistical Descriptions of Spatial Patterns

Recommended Reading

1. Cressie, N.A.C.: Statistics for Spatial Data. Wiley & Sons, New York (1993)
2. Besag, J.: Spatial interactions and the statistical analysis of lattice systems (with discussion). J. Royal Stat. Soc., Series B **36**, 192–236 (1974)
3. Givens, G.H., Hoeting, J.A.: Computational Statistics. Wiley, New Jersey (2005)
4. Lindley, D.V., Smith, A.F.M.: Bayes estimates for the linear model. J. Royal Stat. Soc., Series B **34**, 1–41 (1972)
5. Berliner, L.M.: Hierarchical Bayesian time series models. In: Hanson, K., Silver, R. (eds.) Maximum Entropy and Bayesian Methods, pp. 15–22. Kluwer Academic Publishers, Dordrecht (1996)
6. Wikle, C.K., Berliner, L.M., Cressie, N.: Hierarchical Bayesian space-time models. J. Envr. Ecol. Stat. **5**, 117–154 (1998)
7. Banerjee, S., Carlin, B.P., Gelfand, A.E.: Hierarchical Modeling and Analysis for Spatial Data. Chapman & Hall/CRC, London, Boca Raton (2004)
8. Diggle, P.J., Tawn, J.A., Moyeed, R.A.: Model-based geostatistics (with discussions). Appl. Stat. **47**(3), 299–350 (1998)
9. Wikle, C.K.: Hierarchical Bayesian models for predicting the spread of ecological processes. Ecol. **84**, 1382–1394 (2003)
10. Besag, J., York, J.C., Mollie, A.: Bayesian image restoration, with two applications in spatial statistics (with discussion) Ann. Inst. Stat. Math. **43**, 1–59 (1991)
11. Whittle, P.: On Stationary Processes in the Plane. Biom., **41**(3), 434–449 (1954)
12. Cressie, N., Huang, H.-C.: Classes of nonseparable spatio-temporal stationary covariance functions. J. Am. Stat. Assoc. **94**, 1330–1340 (1999)
13. Stein, M.L.: Space-time covariance functions. J. Am. Stat. Assoc. **100**, 320–321 (2005)

14. Waller, L.A., Xia, B.P., Gelfand, A.E.: Hierarchical spatio-temporal mapping of disease rates. J. Am. Stat. Assoc. **92**, 607–617 (1997)
15. Wikle, C.K., Cressie, N.: A dimension-reduced approach to space-time Kalman filtering. Biom. **86**(4), 815–829 (1999)

Hierarchies

► Hierarchical Spatial Models

Hierarchies and Level of Detail

SABINE TIMPF
Department for Computer Science VI,
University of Würzburg, Würzburg, Germany

Synonyms

Level of detail; LOD; Acyclic directed graph; Partial order; Abstraction; Cognition; Aggregation; Generalizing; Filtering

Definition

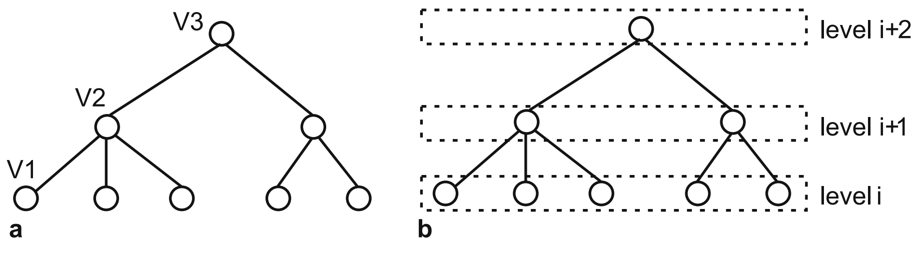
A hierarchy is an ordered structure. Order can be established between individuals or between classes of individuals. The ordering function may be any function defining a partial order:

$$v_1 < v_2 \quad \text{iff} \quad \exists f: v_2 = f\{v_i\}; v_1 \in \{v_i\}\,.$$

A vertex v_1 in a graph is on a lower level than v_2 if and only if there exists a function f, such that v_2 can be calculated by applying f to a set of v_i of which v_1 is a member. The function f must be reflexive, antisymmetric and transitive.
The partial ordering can be depicted as a tree with the vertices denoting the individuals of the domain and the edges representing the ordering function between individuals (Fig. 1a).
The notion of levels is introduced by the idea that vertices at the same depth of the tree belong to the same level of the hierarchy. Thus, there are as many levels in the hierarchy as the tree is deep. The highest level is the most abstracted level, the lowest level is the most detailed level (Fig. 1b).
A hierarchy is intricately linked to the idea of levels. Individuals on the same level share a common property, e. g. individuals have the same power in the company, or map objects have the same level of detail. The order of the levels is always total, although the order between individuals in the hierarchy can only be partial.

Historical Background

Current Geographic Information Science lacks the structures, tools, and operations to handle multiple representations and especially representations with multiple levels of detail. One reason for this is that very little is known about using hierarchies for the description of ordered levels of detail. Hierarchies appear in many different spatial contexts, for example, road networks, political subdivisions, land-use classes, or watersheds. Hierarchies appear also in non-spatial situations such as organizational hierarchies. It is the intrinsic nature of hierarchies that has not yet received much attention and this lies at the basis of the problems with hierarchies in GIScience.
Hierarchies have many advantages usable for GIScience. They provably reduce processing time and increase the stability of any system (Pattee 1973). They break down the task into manageable portions (Timpf et al. 1992; Nardi 1996), enhancing the potential for parallel processing while inferences can be refined depending on the type and level of query. Hierarchies and the principles that build them are interesting as knowledge acquisition and knowledge/data representation methods (Timpf 1999). Hierarchies and their principles are often used as data representation methods. The most prevalent ordering variable in all these hierarchies is scale, i. e., the level of detail of geographic phenomena.
The work with hierarchies is based on the assumption that many individual object-systems with simple behavior form a whole object-system with complex behavior. For example, a map series is a complex object-system composed from many maps. A map in its turn is also composed of many elements. It is also assumed that the organization of the individual objects is hierarchical; the com-



Hierarchies and Level of Detail, Figure 1 **a** Partial ordering: tree; **b** levels in a hierarchy plex object being on top of the hierarchy. This assumption can be found in diverse fields such as systems theory (Koestler 1967; Mesarovic, Macko, and Takahara 1970; Pattee 1973), ecology (Ahl and Allen 1996), biological organisms (Braitenberg 1984), software specification (Brodie, Mylopoulos, and Schmidt 1984), landscape planning (Alexander, Ishikawa, and Silverstein 1977), and reverse engineering.

Hierarchies in Cognitive Science

Hierarchies are fundamental to human cognition (Langacker 1991) and have been studied extensively in the cognitive sciences. Humans arrange information hierarchically and use hierarchical methods for reasoning (Stevens and Coupe 1978). '*Hierarchization*' is one of the major conceptual mechanisms to model the world. Other mechanisms to express structures of the world are, for example, using prototypes, relations, or constraints. The idea is to deduce knowledge at the highest (coarsest) level of detail in order to reduce the amount of facts taken into consideration. Too much detail in facts often means long and inefficient reasoning.

Humans construct hierarchies by abstracting information, that means by building ordered classes of information. For example, when seeing a tree, one recognizes that the brown long trunk in combination with the green leaves fall into the object category 'tree'. In this example, the spatial arrangement of the entity from the class 'leaves' and the entity from the class 'trunk' suggests that the aggregate entity belongs to the class 'tree'. The corresponding hierarchy is an aggregation hierarchy. Hierarchies of entities in the geographic world result in multiple representations within a spatial database.

What makes the conceptual task of modeling hierarchies so difficult is the fact that humans are able to switch between different types of hierarchies (attributes, classes, instances, tasks etc.). Humans do not even notice that they are using different types of hierarchies simultaneously. Timpf (Timpf 1999) assumes that this fact is the biggest impediment to the investigation of representations with multiple levels of detail.

Hierarchies in Systems Theory

In systems theory the observation is made that most biological systems are hierarchically structured and this structure is applied to non-biological systems (Ahl and Allen 1996). This approach was especially taken in the description of complex dynamic systems (Mesarovic, Macko, and Takahara 1970; Pattee 1973).

Simon (Simon 1996) states that complex systems are usually hierarchically structured. There are several advantages to a hierarchical structure. The system has a number of interrelated subsystems that are themselves hierarchically structured. Each subsystem is a stable intermediate form and can function without 'help' from the complex structure. All forms in evolution are stable intermediate forms.

Koestler (Koestler 1967) explains the principle with the parable of the watchmakers. One watchmaker assembles a watch from scratch with all single parts on the table. The other watchmaker assembles smaller subparts of the watch and then connects these subparts. The second watchmaker is much faster in assembling a watch. This is possible in all systems that are nearly decomposable, which means that the relations inside a subsystem are stronger than the interrelations between subsystems.

Mesarovic (Mesarovic, Macko, and Takahara 1970) introduces three types of hierarchical systems distinguished by the types of levels: the level of abstraction, the level of decision complexity and the organizational level. The level of abstraction in a hierarchical system expresses the fact that some objects in the system may be more abstract than others, and the level imposes an order on the objects. The level of decision complexity creates a hierarchy of decision layers, meaning that decisions in a system are broken down to decisions of subsystems. The original decision is made with a certain margin for error, because not all subsystems may be able to decide. The organizational level creates an organizational hierarchy as in a human formal hierarchy.

Scientific Fundamentals

Types of Hierarchies: Overview

Hierarchies are formed through *abstraction mechanisms*, i. e., through factoring out commonalities in the description of several concepts into the description of a more general concept. Four abstraction mechanisms have been identified:

- classification/instantiation
- aggregation/decomposition
- generalization/specialization
- filtering.

The abstraction mechanism of classification is a prerequisite for all other abstraction mechanisms. It is necessary to classify objects before applying operations like aggregate, generalize, or filter to them.

Types of Hierarchies: Aggregating

The type of hierarchy that is constructed by aggregating sets of individuals to singular individuals is called *aggregation hierarchy*. Other names like nested hierarchy, container hierarchy or part-of-hierarchy are also common.

The possibility of aggregation depends solely on a certain attribute of the individuals. For example, containers can be aggregated to form larger containers if they are spatially adjacent. The aggregation hierarchy is the most common type of hierarchy.

$$v_1 < v_2 \quad \text{iff} \quad \exists f_{\text{agg}} : v_2 = f_{\text{agg}}\{v_i\};\ \ v_1 \in \{v_i\}\,.$$

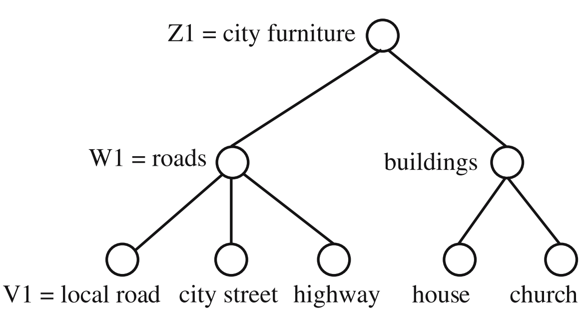
The aggregation function maps a set of individuals to a single individual, or it maps a set of classes of individuals to a single class of individuals. The end of the aggregation is reached if there is only one individual or class of individuals left as argument of the function. The type of the individuals is the same throughout the hierarchy. Only individuals of the same type can be aggregated. Figure 1a shows a structure that represents an aggregation hierarchy.

Types of Hierarchies: Generalizing

The *generalization hierarchy* defines classes as more generic the higher the level of the class in the hierarchy. E. g., houses and industrial buildings all belong to the generic class 'building', or local road, city streets, and highways all belong to the generic class 'roads' (see Fig. 2). This type of hierarchy can also be called classification hierarchy, taxonomy or is-a-hierarchy (because it states that any entity in a subclass *is a*n entity in the superclass as well.

The generalization hierarchy relates a class to a superclass. V1, W1, and Z1 denote classes in Fig. 2. Generalization is defined as forming a new concept by leaving out properties to an existing concept. The inverse function of specialization forms a new concept by adding properties to an existing concept. This hierarchy is a non-nested hierarchy in contrast to the aggregation hierarchy, which is a nested hierarchy.

$$V_1 < W_1 \quad \text{iff} \quad \exists f_{\text{gen}} : W_1 = f_{\text{gen}} V_1$$



Hierarchies and Level of Detail, Figure 2 Visualization of a generalization hierarchy

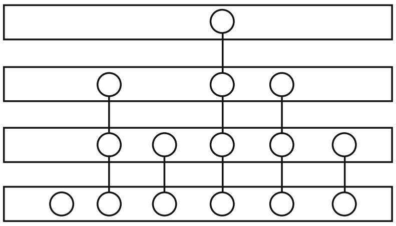
The rules, determining what class belongs to what generic class, are predetermined. The generalization function states explicitly what classes generalize to what generic classes. This can be visualized as a hierarchy of classes with the most generic classes at the top level. Figure 2 shows a generalization hierarchy with a single most generic class Z1 at the top level. Individuals change their type when generalized, e. g., individuals of class V1 become individuals of class W1 after generalization. Notice that the structure of the hierarchy is the same as the structure of the aggregation hierarchy (see Figs. 1a and 2), but that the function producing the hierarchies is quite different.

Types of Hierarchies: Filtering

The *filter hierarchy* applies a filter function to a set of individuals on one level and generates a subset of these individuals on a higher level. The individuals at the higher level are also represented at the lowest level. Individuals pass the filter at one or more levels of detail. Individuals that do not pass the filter disappear from the representation (Fig. 3). This is the most striking difference between the two other hierarchies and this hierarchy. The class and the type of the individuals stay the same.

$$s_1 < s_2 \quad \text{iff} \quad \exists f_{\text{fil}} : s_2 = f_{\text{fil}}(s_1)\,.$$

Two sets of individuals s_1 and s_2 are partially ordered if there exists a function f_{fil} such that the set s_2 can be filtered from the set s_1, i. e., such that s_2 is a subset of s_1.



Hierarchies and Level of Detail, Figure 3 Filter hierarchy

Key Applications

Knowledge engineering, software engineering, database theory, object-oriented programming, database generalization (levels of detail), computer graphics (levels of detail)

Future Directions

Combinations of hierarchy types and their mathematical properties, implementations of partially ordered sets in efficient data structures, transformations between hierarchies especially between differently typed hierarchies

Cross References

- ► Abstraction of GeoDatabases
- ► Approximation
- ► Cadastre
- ► Geospatial Semantic Web
- ► Hierarchical Spatial Models
- ► Knowledge Representation, Spatial
- ► Map Generalization

Recommended Reading

Ahl, V., Allen, T.F.H.: Hierarchy Theory – A Vision, Vocabulary, and Epistemology. Columbia University Press, New York (1996)

Alexander, C., Ishikawa, S., Silverstein, M.: A Pattern Language – Towns, Buildings, Construction. Oxford University Press, New York (1977)

Braitenberg, V.: Vehicles, experiments in synthetic psychology. MIT Press, Cambridge, MA (1984)

Brodie, M., Mylopoulos, J., Schmidt, J.: On Conceptual Modelling. Topics in Information Systems. Springer-Verlag, New York/Heidelberg (1984)

Koestler, A.: The Ghost in the Machine. Pan, London (1967)

Langacker, R.W.: Foundations of Cognitive Grammar – Theoretical Prerequisites, 1. Stanford University Press, Stanford, CA (1991), 516 pp

Mesarovic, M.D., Macko, D., Takahara, Y.: Theory of hierarchical, multilevel, systems. Mathematics in science and engineering. Academic Press, New York (1970)

Nardi, B. (ed.): Context and consciousness – activity theory and human-computer interaction. The MIT Press, Cambridge, MA (1996)

Pattee, H.H. (ed.): Hierarchy Theory – The Challenge of Complex Systems. Braziller, New York (1973) 156 pp

Simon, H.A.: The Sciences of the Artificial. MIT Press, Cambridge, MA (1996)

Stevens, A., Coupe, P.: Distortions in Judged Spatial Relations 10. In: Cognitive Psychology **10**(4), 422–437 (1978)

Timpf, S., Volta, G.S., Pollock, D.W., Egenhofer, M.J.: A Conceptual Model of Wayfinding Using Multiple Levels of Abstractions. In: Frank, A.U., Campari, I., Formentini, U. (eds.) Theories and Methods of Spatio-temporal Reasoning in Geographic Space. Lecture Notes in Computer Science 639, pp. 348–367. Springer Verlag, Heidelberg-Berlin (1992)

Timpf, S.: Abstraction, levels of detail, and hierarchies in map series. In: Freksa, C., Mark, D.M. (eds.) Spatial Information Theory – cognitive and computational foundations of geographic information science (COSIT'99). Lecture Notes in Computer Science 1661, pp. 125–140. Springer-Verlag, Berlin-Heidelberg (1999)

High-Level Features

► Feature Extraction, Abstract

Homeland Security and Spatial Data Mining

VANDANA P. JANEJA[1], NABIL R. ADAM[2]

[1] Information Systems Department, University of Maryland, Baltimore, MD, USA

[2] Management Science and Information Systems Department, Center for Information Management, Integration and Connectivity, Rutgers University, Newark, NJ, USA

Synonyms

Neighborhood; Scan statistic; Outlier; Spatial data mining; Anomaly detection

Definition

Spatial data mining deals with the discovery of non trivial patterns in spatial datasets. It includes the discovery of regularly occurring patterns such as associations between objects and the abnormal patterns such as outlying objects in the data. This discovery differs from traditional data mining due to the complex nature of spatial data due to two important properties of *spatial autocorrelation* such that nearby objects behave in a similar manner to each other and *spatial heterogeneity* where nearby objects may exhibit different properties. This entry outlines the discovery of spatial anomalies which are abnormal patterns in the data. This is pertinent to the discovery for homeland security where peculiar objects and associations in the spatial context may be highly relevant. An overview is presented of spatial data mining techniques which play a key role in homeland security applications, specifically providing insight into (1) Neighborhood based anomaly detection techniques and (2) Scan Statistics based approaches. Subsequently, several Homeland Security applications benefiting from these techniques are outlined.

Historical Background

Events such as the September 11, 2001, and terrorist attacks on the World Trade Center and the Pentagon have regional and national repercussions. These events have brought the issues of homeland security in sharp focus. In the event of an emergency (natural or man-made), the chaos at the epicenter of a disaster calls for immediate and accurate response to events and communications between first responders and emergency management organizations. Regional organizations are beginning to fill this niche by promoting efforts to coordinate and manage information sharing (such as [15,16]). Various situations may emerge which can challenge the preparedness of various governmental agencies. Any homeland security scenario brings about the need for resource management and coordination of various tasks. This can be facilitated by various spatial data mining techniques.

First, a motivating example is presented to illustrate such a need in the domain of homeland security. Let us consider the following scenario:

A malevolent entity is preparing to ship toxic liquid organic NOS, a poisonous chemical, into the United States. The same a cargo shipment originates in North Africa, however, the shipping manifest shows the country of origin as Spain. It crosses the border at Newark, New Jersey (one of the biggest ports in the eastern border). Due to the benign information gathered from the cargo manifest and the established reputation of the country of origin and the shipping company, this shipment leaves the inspection facility unchecked. This cargo container transfers the material into a single unit vehicle with a length of less than 45 feet, with outside overall body width of less than 8 feet and overall height less than 12 feet. The container makes its way toward New York city via the Lincoln Tunnel. Since the container size is within the limit set by the authorities inspecting the vehicles, the shipment proceeds into the tunnel. A spill is caused within the tunnel by the malevolent entities. The sensors located in the tunnel start sending warnings to the Emergency Operations Center (EOC). The port authority issues general alerts to respective agencies and local law enforcement agencies are notified. Measures need to be taken to contain the toxic spill and its effect on the nearby vicinity. The authorities need to perform a risk assessment to identify how the wind may carry the chemical to immediate risk areas. They also need to identify evacuation facilities and hospitals in the potentially affected areas. At the same time, pharmacies need to be monitored for a sudden rise in the purchase of medicines that may be linked to exposure of individuals to the harmful chemical fumes.

The above scenario brings out various situations where homeland security may benefit from Spatial data mining techniques. Specifically, (i) identifying wayward cargo shipments based on geospatial associations, (ii) Sensor monitoring applications to distinguish between benign and non-benign situations, (iii) Disease surveillance in the local affected area, (iv) Alert management for anomalous scenarios in emergency management, and (v) Management of resources located in the area to coordinate between the various agencies. Each of these representative applications proposed to address the needs for homeland security deriving from various spatial data mining techniques is discussed in the following sections.

Scientific Fundamentals

1. Neighborhood Based Anomaly Detection

Spatial anomaly detection techniques account for two important properties in the data, namely, spatial autocorrelation and heterogeneity. Spatial objects behave in a similar manner by virtue of being in proximity to each other, thereby depicting autocorrelation. However certain underlying causes (such as presence of a landfill, mountain range etc.) may make these objects behave differently from each other even though they are close to each other, thereby demonstrating heterogeneity. It is essential to account for these two properties for any spatial data mining technique. First, anomaly detection applications, which are devised based on the spatial neighborhood formation primarily taking into consideration autocorrelation, are presented. Subsequently, two techniques which account for both autocorrelation and heterogeneity are discussed. Specifically, (a) Spatial outlier detection in autocorrelated neighborhoods and (b) Discovery in heterogeneous neighborhoods, namely, (i) Trajectory analysis and (ii) Spatial Outlier detection in heterogeneous neighborhoods.

a. Spatial Outlier Detection in Autocorrelated Neighborhoods The issue of graph based spatial outlier detection using a single attribute has been addressed in [17]. Their definition of a neighborhood is similar to the definition of a neighborhood graph as in [4], which is primarily based on spatial relationships. However the process of selecting the spatial predicates and identifying the spatial relationship could be an intricate process in itself. Moreover, the definition of a spatial neighborhood does not capture the semantic relationship between the attributes of the spatial objects and their respective areas of influence. Further, the outlier detection is performed based on a single attribute in the neighborhood. Similarly, Spatial Local Outlier Measure (*SLOM*) [18] uses a traditional notion of spatial neighborhood to identify local outliers. However, it suppresses the reporting of outliers in highly unstable areas where data is too heterogeneous. [10] proposed an approach to discover spatial outliers with multiple attributes using Mahalanobis distance to analyze spatial data with multiple attributes, considering the average and median of the neighborhood values to determine the deviation of a spatial object from the normal behavior in the neighborhood. [8] proposed a technique for weighted spatial outlier detection. They detect the outliers in spatial neighborhoods which are weighted based on the impact of spatial attributes such as location, area, and contour. However, the outlier detection is performed based on a single attribute in such a neighborhood. A clustering technique using Delaunay triangulation (*DT*) for spatial clustering is proposed in [7]. It identifies outliers as a by-product of clustering. It connects the points by edges if they are within a certain threshold proximity. Essentially, in all of the above approaches the spatial neighborhood does not account for spatial autocorrelation and heterogeneity in combination.

b. Discovery in Heterogeneous Neighborhoods Each spatial object (such as a sensor or a city) is associated with spatial and non-spatial attributes. The neighborhood formation[12] first generates the immediate neighborhood of an object, which is called the *micro neighborhood*. In addition to the spatial and non-spatial attributes of the object in it, a micro neighborhood captures all the implicit knowledge about the immediate neighborhood of this object, such as the presence or absence of features (e. g., landfill) in its proximity. It identifies *spatial relationships* (such as adjacency) using the spatial attributes of the micro neighborhoods to account for autocorrelation. In order to capture heterogeneity, it identifies *semantic relationships* (such as similarity) using the non-spatial attributes and features in the micro neighborhoods. It then uses both spatial and semantic relationships to merge appropriate micro neighborhoods into larger neighborhoods called as *macro neighborhoods*. Thus, the formed macro neighborhood captures both autocorrelation and heterogeneity in the neighborhood definition. Subsequently, anomaly detection is performed in these neighborhoods. Two specific types of anomaly detection techniques are described which utilize this neighborhood information for (i) identifying anomalous trajectories (such as cargo routes) and (ii) identifying spatial outliers.

(i) Geospatial Trajectory Analysis A geospatial trajectory represents a connected path of a series of discrete points that have spatial coordinates. The geospatial points forming the trajectories are associated with data that can be both spatial and non-spatial pertaining to different domains of interest. An anomalous geospatial trajectory is one that deviates from the norm.

The approach of detecting anomalous geospatial trajectories begins by considering the smallest geospatial entities, called spatial units. As an example, a spatial unit could be a city associated with spatial coordinates and other spatial and non-spatial information including economic data, agricultural data, boundary data (demarcations of borders or a country), etc., whose sources are from various domains. A macro neighborhood is generated by merging micro neighborhoods formed around these spatial units. Essentially, this forms a characterization of the region in order to generate a larger spatial entity from the spatial units that are spatially related and are similarly behaving with respect to their attributes. The characterized region is the input for the detection of anomalous geospatial trajectories. A composite feature vector for each macro neighborhood is generated and the associations (called path associations) are identified between the macro neighborhood and a micro neighborhood on a geospatial trajectory. This continues in a drill down manner to identify geospatial trajectory that has a path association with micro neighborhoods that are not a part of the trajectory, thus indicating possible anomalous behavior.

(ii) Spatial Outlier Detection Often, it is required to perform outlier detection in spatial datasets (e. g., in environmental monitoring, transportation with RFID tags, monitoring of subways and tunnels, etc.) to identify different types of anomalous behavior. Here, a spatial object is associated with spatio-temporal points monitoring a phenomenon over time at a certain spatial location (for example, a sensor of a spatial object and the spatio-temporal readings of a sensor are the points associated with the spatial object). The objects may be placed at different positions in a region. In such a setup, the challenge lies in identifying abnormal objects and abnormal points traversing a section of the region. Once the macro neighborhoods are identified, anomaly detection is performed in them such that the outlier is detected with respect to the neighborhood. Three types of outliers are identified, namely, *Spatio-temporal outliers*, *Spatial outliers* and *Temporal outliers*. Spatio-temporal outliers are the anomalous points which are at a large distance from the other points in the neighborhood. Thus, they do not conform to the general behavior of the neighborhood (for instance, anomalous readings of a sensor). A spatial outlier is the object which has a high number of spatio-temporal outliers associated with it. Thus, it is abnormal as compared to other spatial objects in the neighborhood in terms of the high number of spatio-temporal outliers it is generating. Furthermore, a couple of spatio-temporal outliers may create a temporal link between two micro neighborhoods by virtue of the small time lag between them. This results in a temporal outlier which indicates an anomaly traversing the neighborhood (such as a toxic spill or chemical fumes traversing a region).

2. Scan Statistic Based Techniques

Scan statistics [13] proposes an intuitive framework for identification of unusual groupings in the data. The idea behind the scan statistic approach is to test if a process (such as the spread of a disease in a region) is purely random, or if any unusual groupings in the form of a scan window can be detected, indicating a non-random cause. The following approaches are discussed (a) Traditional Scan Statistics approaches, (b) Circle based Spatial Scan Statistic [9], (c) FlexScan [19] and (d) Free-Form Spatial Scan Statistic approach [5].

a. Traditional Scan Statistics Approaches LS^3 approach [6] extends the scan statistic approach proposed in [13]. It proposes a linear semantic based scan statistic (LS^3) which accounts for variability in the data for identifying windows over linear paths, taking into account autocorrelation and heterogeneity. However, this approach is limited to linear paths and linear windows, and is not generalized for spatial regions and irregular shaped windows.

Openshaw [14] has proposed an approach using a regular grid superimposed over the region where circles of varying radii are drawn on the grid nodes. The circles attaining a certain level of significance are flagged as the unusual groupings. Besag et al. [1] fix the number of cases within a circle and then search for those with a small risk population that makes them highly significant. Most of these approaches are limited to circular windows which do not capture the natural phenomenon. The simulated annealing [3] approach considers a set of polygons and identifies the most unusual groupings of these polygons using maximum likelihood as a measure. Although this results in irregular shaped windows, it does not account for spatial heterogeneity in the data since the grid nodes are not grouped based on data attributes, but purely based on being part of a circle. Thus, most of the scan statistic based approaches suffer from two major issues i) the window shape is not free form, and ii) the anomaly detection is primarily based on autocorrelation only and does not take into account heterogeneity.

b. Circle Based Spatial Scan Statistic For spatial datasets, a variation of the one-dimensional scan statistic, called spatial scan statistic, has been proposed [9]. It is the most widely used and extended approach. In this approach, the spatial region is scanned with a circular window of varying size. By searching for windows without specifying the size or location, this technique is able to avoid pre-selection biases. It provides a measure of whether the observed number of objects is unlikely for a window of a certain size using reference values from the entire study region. A spatial scan statistic is useful in several applications where the relationships may not be necessarily linear, for instance, in identifying the disease clusters over a region which may not travel in a linear manner. However, such a spatial scan statistic suffers from two limitations. First, it does not identify irregularly shaped windows, however, natural phenomenon, such as the spreading of disease, seldom take up a fixed shape such as a circle. Secondly, it does not take heterogeneity into consideration, thus, spatial objects which may be part of a circle shaped window may not necessarily have anomalous behavior, but by virtue of proximity to anomalous entities, may be considered as a part of the anomalous window.

c. FlexScan The FlexScan approach identifies flexible shaped scan windows [19]. It identifies a window with K number of points where K varies from 1 to a pre-set maximum K which are adjacent to each other. Here, the regions in the window are a subset of the regions of starting spatial location i and its $K-1$ nearest neighbors. However, this approach predefines K and only considers adjacency as the factor for moving to the next location in the dataset, essentially overlooking heterogeneity.

d. Free-Form Spatial Scan Statistic The FS^3 (**F**ree-Form **S**patial **S**can **S**tatistic) approach [5] takes both heterogeneity and autocorrelation into account, and identifies free-form anomalous spatial windows that are not necessarily of traditional shapes. Such a window will consist of a contiguous region represented by a subset of spatial objects.

This approach first constructs a weighted Delaunay nearest neighbor graph (*WDNN*). Essentially, an edge between two spatial objects exists if autocorrelation identified in some sort of a spatial relationship exists. The weight on the edge reflects the heterogeneity of the attributes among the spatial objects. The spatial heterogeneity among the features associated with the spatial objects is captured by adding weights to the *DNN*. The weight for each edge is computed based on the set of attributes associated with each spatial object. The weighted *DNN* (*WDNN*) graph is used to determine the transition strategy which is subsequently employed to scan the region using a *random walk*. Thus, unlike the circle based scan statistic where the transition is automatic by virtue of the object being in the circle, here, the transition will be based on the relationships between the attributes quantified using the weight on the edges of the graph. The transition relationships are defined in terms of a transition probability from one object to the other. The weight for each edge, called the *transition probability weight*, is computed based on the set of attributes associated with each spatial object.

The above constructed graph is traversed using random walks. The result of the random walk is a free-form window which eliminates the restriction of a fixed shape scan window for the purpose of conducting a scan statistic to determine if it is anomalous or not. A scan statistic, in turn, employs Monte Carlo simulations to attach a p-value with the result of determining that the window identified in the previous step is indeed anomalous and not merely occurring randomly.

Key Applications

In the following section, various applications for the homeland security domain which benefit from the spatial data mining techniques are discussed.

1. Cargo Route Analysis

Based on the motivating scenario, consider the case of an anomalous shipment which originates from one country but depicts another as its origin on the manifest. Such a problem is termed as a transshipment in customs terminology. Transshipment, in general, refers to the movement of goods through multiple stopovers en-route to a destination. While transshipment is legal and is commonly used by businesses, often it is exploited for the purpose of circumventing trade laws and restrictions applicable to the shipment, and to ship unlawful merchandise such as contaminated or prohibited foods, illegal drugs, weapons, and hazardous wastes.

Since a cargo shipment follows a specific route, one may exploit the spatial and non-spatial attributes associated with each geographic location in the route. A particular cargo can be strongly associated with some part of the route than other parts. Alternatively, it could be strongly associated with some geographic region that is not on the route which would indicate the possibility of the transshipment prompting further investigation. Given a cargo route, it is first divided into segments where each segment is associated with its feature vector. The composite feature vector of the characterized region and the feature vector of the segments are used to identify associations which could be either expected or unexpected. If the segment is most strongly associated with its own region, then it validates the shipment route. However, any strong association between a characterized region and a segment that does not reside in the region indicates a potential transshipment. For example, if one of the attributes of the cargo route segment is related to drug activity, then all the regions that are characterized as active drug zones should be taken into consideration for further investigation to identify potential relations.

2. Sensor Monitoring

This application aims to identify benign versus non-benign scenarios for various environmental, radiological and biological sensors. For instance, (i) Environmental scientists may want to identify abnormally behaving water monitoring sensors, (ii) A customs agent may want to discover anomalies among cargo shipments with RFID tags to identify potentially deviant shipments even before they cross the border, or (iii) City officials may want to identify threats or malfunctions based on numerous sensors placed around a metropolitan area in subways and tunnels, etc. Consider anomaly detection for toxicity levels in a stream where various spatial objects, namely, sensors, are placed. Each sensor measures a certain attribute, namely, the toxicity level. At first, the neighborhood is formed mainly based on spatial autocorrelation such that nearby sensors are merged in the neighborhood by virtue of proximity. Thus, if two spatial objects are adjacent or have an edge connecting them, they may form a single spatial neighborhood. In this neighborhood formation, the values of the toxicity levels or other underlying features are not considered. As a result, if one sensor is placed in a different tributary which is under the influence of a nearby land fill, it may not be detected. Thus, some extreme values may be considered to be part of the neighborhood mainly based on spatial autocorrelation even though they may be part of some other semantically related neighborhood. This will lead to identification of frivolous anomalies. However, if the autocorrelation captured by adjacency relationships and heterogeneity captured by the underlying features, spatial and non spatial attributes is considered, then it can be observed that the neighborhoods may be much more refined. Therefore this approach associates as much semantic knowledge as possible to produce a complete spatial characterization of a region represented in a spatial neighborhood such that subsequent knowledge discovery may be performed with more accurate results.

3. Disease Surveillance

Anomalous windows are the unusual groupings of contiguous points that might occur in the data. Detection of anomalous windows is essential for the identification of unusual numbers of disease cases. This is critical to the domain of homeland security as it might indicate a bio terrorism threat. If two cities are adjoined (spatially), due to the autocorrelation, one can expect a similar disease spread behavior in these two cities. On the other hand, if they are separated by a mountain range, even if they are adjoining, they may behave differently. This aspect is accounted for by spatial heterogeneity. Thus, disease surveillance applications need to consider these aspects carefully in order to accurately determine the extent of impact of a threat and also the spatial extent for disease surveillance.

4. Alert Management

consists of various aspects including management at the local law enforcement level and communication at multiple agency levels. These aspects are discussed here.

As can be seen from the motivating example presented, the following three different requirements emerge: 1) There is an overwhelming number of alarms generated by different agencies, people and sensors which often cause confusion and are difficult to interpret when considered in this unfiltered, non-aggregated form, 2) There could be many false positives in this stream of alarms and it is critical for high level decision making to determine the true alerts vs. false positives, and 3) For the decision makers, it may be required to associate an alert with a dynamic flow of the alert over a period of time, thus, there is a need to integrate the alerts with certain models to interpret and project the alert scenario in a dynamic setting to help in making more informed decisions. The alert generation for such a complex system would involve the following steps. 1) Alarm preprocessing: The objects are characterized using the neighborhood definition as discussed in section "1. Neighborhood based anomaly detection". By identifying spatial outliers(such as entirely malfunctioning sources), false positives can be identified in the streams of alarms. 2) Aggregated Alert Generation: The series of alarms are interpreted based on different relationships between the alarms which will lead to an aggregation of alarms into alerts. Aggregated alerts are those alerts which are identified based on a clustering of alarms. If a cluster of alarms has a significant number of alarms from a single or multiple source(s), then this cluster may be termed an aggregated alarm. Secondly, if there is a set of clusters in which no one cluster has a significant number of alarms, then it looks for a mutual overlap of the clusters and this overlap becomes the aggregated alert. 3) Dynamic flow alert generation: The alert generated will be integrated with a relevant model for generating the dynamic flow of the alert based on some factors determined by the external model. Based on the input of a model, it identifies the dynamic flow of an alert starting from a source and flowing in a certain direction given the time lag and direction of the flow of the alert. Thus, this system generates aggregated alerts from a series of alarms.

5. GIS for Local Law Enforcement

This application has been successfully employed to visualize crime data to look for suspicious correlations and activities [20,21]. COPLINK is a tool for local law enforcement and [20], among other components, consists of a spatio-temporal visualization tool for crime analysis which acts as an interface to COPLINK. It integrates a periodic view, a timeline view, and a GIS view where the periodic visualization displays patterns with respect to time. Timeline visualization displays a linear visualization of the temporal data and the GIS visualization displays information on a map. This interface enhances COPLINK by providing an interactive environment to access and analyze police data dynamically.

6. GeoCollaborative Crisis Management

Crisis management is a critical need at the time of a natural or man made disaster. It relies upon collaboration of people for extracting the necessary knowledge from geospatial data. The activities succeeding a disaster are coordinated based on such knowledge. Current emergency management approaches fail to support group work and are typically designed without integrating the knowledge of group interaction at the time of crisis. [2,11] have proposed an approach to geocollaborative crisis management which addresses these limitations. First, it facilitates group work in Emergency Operation Centers (EOCs) with GIS enabled large screen displays using multimodal, gesture-speech interfaces. Secondly, it supports distributed teams with multimodal devices such as mobile devices in the field linked to others using desktop or large-screen displays in the EOC or in mobile field stations.

Future Directions

This entry has reviewed the various spatial data mining techniques and applications for Homeland security. There is a need for a unifying system which brings these applications under a single homeland security application together, providing a seamless flow of data into the system followed by geospatial data mining. The outcome can be presented as knowledge to the frontline officers and decision makers. Homeland security domains require strict adherence to privacy, however, this issue has not been fully explored for spatial data mining techniques. Moreover, since these applications require interfacing with several disparate systems and large datasets, efficiency issues for spatial data mining techniques need to be addressed.

Cross References

► Biomedical Data Mining, Spatial
► Gaussian Process Models in Spatial Data Mining

Recommended Reading

1. Besag J., Newell, J.: The detection of clusters in rare diseases. J. R. Stat. Soc. Series A **154**, 143–155 (1991)
2. Cai, G., Bolelli, L., MacEachren, A.M., Sharma, R., Fuhrmann, S., McNeese, M.: Geocollaborative crisis management: using maps to mediate eoc–mobile team collaboration. In: dg.o '04: Proceedings of the 2004 annual national conference on Digital government research, pp. 1–2. Digital Government Research Center (2004)

3. Duczmal L., Renato, A.: A simulated annealing strategy for the detection of arbitrarily shaped spatial clusters. Comput. Statist. Data Anal. **45**(2), (2004)
4. Ester, M., Kriegel, H., Sander, J.: Spatial data mining: A database approach. In: 5th International Symposium on Advances in Spatial Databases, pp 47–66. Springer-Verlag, London, UK (1997)
5. Janeja V.P., Atluri, V.: FS^3: A random walk based free-form spatial scan statistic for anomalous window detection. In: ICDM, pp. 661–664 (2005)
6. Janeja V.P., Atluri, V.: ls^3: A linear semantic scan statistic technique for detecting anomalous windows. In: ACM Symposium on Applied Computing (2005)
7. Kang, I., Kim, T., Li, K.: A spatial data mining method by delaunay triangulation. In: 5th ACM int. workshop on Advances in GIS, pp. 35–39 (1997)
8. Kou, Y., Lu, C., D. Chen, D.: Spatial weighted outlier detection. In: Proceedings of the 6th SIAM Int. Conference on Data Mining 2006, Bethesda, MD, USA, 20–22 April (2006)
9. Kulldorff, M.: A spatial scan statistic. Commun. Stat. – Theory Meth. **26**(6),1481–1496 (1997)
10. Lu, C., Chen, D., Kou, Y.: Detecting spatial outliers with multiple attributes. In: 15th IEEE International Conference on Tools with Artifcial Intelligence, p. 122 (2003)
11. MacEachren, A.M., Cai, G., McNeese, M., Sharma, R., Fuhrmann, S.: Geocollaborative crisis management: Designing technologies to meet real-world needs. In: Proceedings of the 7th annual national conference on Digital government research. Digital Government Research Center (2006)
12. Adam, N., Janeja, V.P., Atluri, V.: Neighborhood based detection of anomalies in high dimensional spatio-temporal sensor datasets. In: 19th ACM Symp. on Applied Computing (2004)
13. Naus, J.: The distribution of the size of the maximum cluster of points on the line. J. Am. Stat. Assoc. **60**, 532–538 (1965)
14. Openshaw, S.: A mark 1 geographical analysis machine for the automated analysis of point data sets. Int. J. GIS **1**(4):335–358 (1987)
15. Port Authority of NY and NJ. http://www.panynj.gov. Accessed May 2005
16. Kenyon, S.H.: Regional Effort Forges Emergency Coordination System, Signal Magazine (2004). http://www.afcea.org/signal/articles/anmviewer.asp?a=13z=2. Accessed Aug 2007
17. Shekhar, S., Lu, C., Zhang, P.: Detecting graph-based spatial outliers: algorithms and applications (a summary of results). In: 7th ACM international conference on Knowledge discovery and data mining, pp. 371–376 (2001)
18. Sun P., Chawla, S.: On local spatial outliers. In: 4th IEEE International Conference on Data Mining, pp. 209–216 (2004)
19. Tango T., Takahashi, K.: flexscan: Software for the flexible scan statistics, version 1.1.2. http://www.niph.go.jp/soshiki/gijutsu/download/exscan/index.html (2005)
20. Wang, G., Chen, H., Atabakhsh, H.: Automatically detecting deceptive criminal identities. Commun. ACM **47**(3):70–76 (2004)
21. Xu J., Chen, H.: Untangling criminal networks: A case study. In: ISI, pp. 232–248 (2003)

Hotspot

► CrimeStat: A Spatial Statistical Program for the Analysis of Crime Incidents

Hotspot Detection and Prioritization

► Hotspot Detection, Prioritization, and Security

Hotspot Detection, Prioritization, and Security

Ganapati P. Patil[1], Raj Acharya[2], Wayne Myers[3], Shashi Phoha[4], Rajan Zambre[5]
[1] Center for Statistical Ecology and Environmental Statistics, Department of Statistics, The Pennsylvania State University, University Park, PA, USA
[2] Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA, USA
[3] School of Forest Resources and Office for Remote Sensing and Spatial Information Resources, Penn State Institutes of Environment, The Pennsylvania State University, University Park, PA, USA
[4] Division of Information Science and Technology, Applied Research Laboratory, The Pennsylvania State University, University Park, PA, USA
[5] Erallo Technologies Inc. Headquarters, Littleton, MA, USA

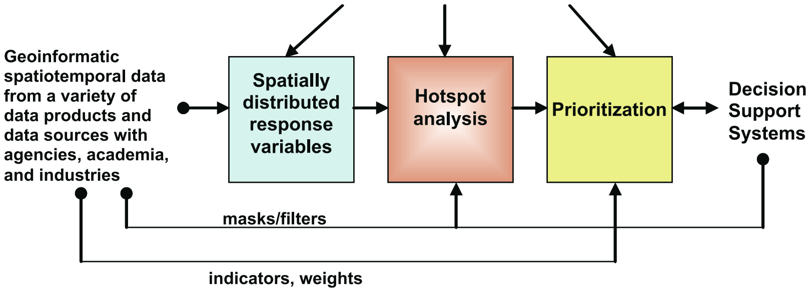
Synonyms

Geoinformatic surveillance; Hotspot detection and prioritization; Early warning; Crime mapping; Mapping and analysis for public safety

Definition

Building on important developments in spatial analysis and fundamental advances in statistical research, and catalyzed by the broad outreach of the Penn State Center for Statistical Ecology and Environmental Statistics, a diverse partnership has gathered to develop cross-cutting solutions that address issues common to multiple national applications. The focal point is a toolkit for geoinformatic hotspot detection and prioritization that provides a new way to rigorously use the geospatial and spatiotemporal information that is available in a variety of national data products on geographic regions and infrastructure networks. The major information flows may be represented schematically as in Fig. 1.

This fascinating work is in progress. The hotspot geoinformatics forum is coming together within the worldwide context of contemporary digital governance. The proposed effort addresses declared needs for geoinformatic surveillance, software infrastructure, and a decision support system for spatial and spatiotemporal hotspot detec-



Hotspot Detection, Prioritization, and Security, Figure 1 A schematic diagram for hotspot geoinformatics decision support-system

tion and prioritization. Hotspot means something unusual: an anomaly, aberration, outbreak, elevated cluster, critical area, etc. The declared need may be for monitoring, etiology, management, or early warning. Responsible factors may be natural, accidental, or intentional.

This multiyear effort will develop methods and tools for hotspot detection and prioritization across geographic regions and across networks. It will apply, adapt, and validate methods and tools for national applications, such as public health, ecohealth, ecosystem condition, watershed management, invasive species, carbon sources and sinks, networked infrastructure security, and object recognition and tracking, among others, leading to a sophisticated next-generation analytical and computational decision support system, beyond the present day health-area-based circular spatial scan. The innovation employs the notion of the upper level set (ULS), and is accordingly called the ULS scan statistic system [1,2]. The proposed effort will also apply, adapt, and validate a prioritization innovation that provides capability for prioritization and ranking of detected hotspots based on multiple criteria indicators without integration of indicators into an index. It employs Hasse diagrams for visualization, and partially ordered sets and Markov chain Monte Carlo (MCMC) methods for analysis [3,4].

Historical Background

In geospatial and spatiotemporal surveillance, it is important to determine whether any variation observed may reasonably be due to chance or not. This can be done using tests for spatial randomness, adjusting for the uneven geographical population density, as well as for age and other known risk factors. One such test is the spatial scan statistic, which is used for the detection and evaluation of local clusters or hotspot areas. This method is now in common use by various governmental health agencies, including the National Institutes of Health, the Centers for Disease Control and Prevention, and the state health departments in New York, Connecticut, Texas, Washington, Maryland California, and New Jersey. Other test statistics are more global in nature, evaluating whether there is clustering in general throughout the map, without pinpointing the specific location of high or low incidence or mortality areas.

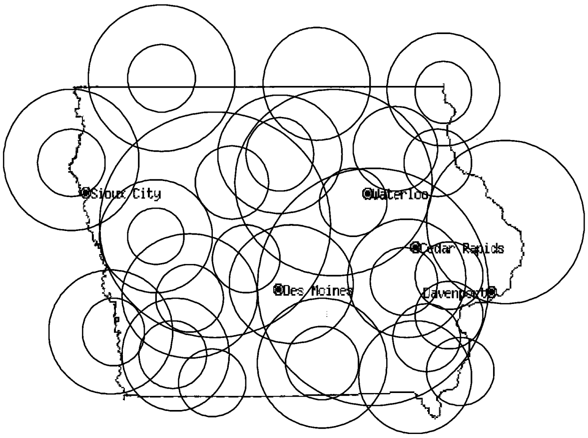
A declared purpose of digital governance is to empower the public with information access, analysis, and policy to enable transparency, accuracy, and efficiency for societal good at large. Hotspot detection and prioritization become natural undertakings as a result of the information access over space and time. Hotspot means spot that is hot, which is of special interest or concern. Geoinformatic surveillance for spatial and temporal hotspot detection and prioritization is crucial in the twenty-first century.

Scientific Fundamentals

Scan Statistic Methodology and Technology

Three central problems arise in geographical surveillance for a spatially distributed response variable. These are (1) identification of areas having exceptionally high (or low) response, (2) determination of whether the elevated response can be attributed to chance variation (false alarm) or is statistically significant, and (3) assessment of explanatory factors that may account for the elevated response. The spatial scan statistic [5] has become a popular method for detection and evaluation of disease clusters. In space-time, the scan statistic can provide early warning of disease outbreaks and can monitor their spatial spread.

Spatial Scan Statistic Background The spatial scan statistic deals with the following situation. A region R of Euclidian space is tessellated or subdivided into cells that will be labeled by the symbol a. Data is available in the form of a count Y_a (non-negative integer) on each cell a. In addition, a "size" value A_a is associated with each cell a. The cell sizes A_a are regarded as known and fixed, while the cell counts Y_a are random variables. In the disease setting, the response Y_a is the number of diseased individuals within the cell and the size A_a is the total number of individuals in the cell. Generally, however, the size variable is adjusted for factors such as age, gender, environmen-



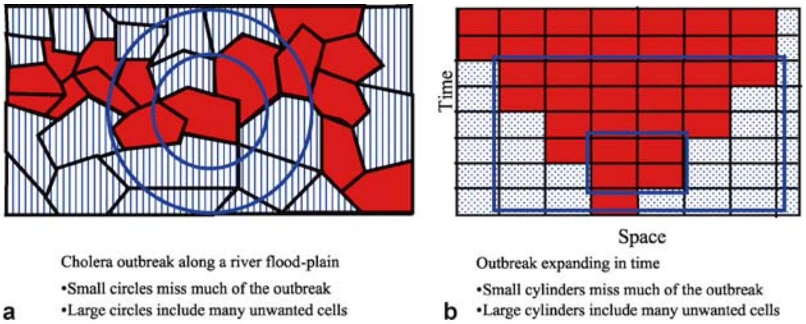
Hotspot Detection, Prioritization, and Security, Figure 2 A small sample of the circles used

tal exposures, etc., that might affect incidence of the disease. The disease rate within the cell is the ratio Y_a/A_a. The spatial scan statistic seeks to identify "hotspots" or clusters of cells that have an elevated rate compared with the rest of the region, and to evaluate the statistical significance (p-value) of each identified hotspot. These goals are accomplished by setting up a formal hypothesis-testing model for a hotspot. The null hypothesis asserts that there is no hotspot, i. e., that all cells have (statistically) the same rate. The alternative states that there is a cluster Z such that the rate for cells in Z is higher than for cells outside Z. An essential point is that the cluster Z is an unknown parameter that has to be estimated. Likelihood methods are employed for both the estimation and significance testing. Candidate clusters for Z are referred to as zones. Ideally, maximization of the likelihood should search across all possible zones, but their number is generally too large for practical implementation. Various devices (e. g., expanding circles as in Fig. 2) are employed to reduce the list of candidate zones to manageable proportions. Significance testing for the spatial scan statistic employs the likelihood ratio test; however, the standard chi-squared distribution cannot be used as reference or null distribution, in part because the zonal parameter Z is discrete. Accordingly, Monte Carlo simulation is used to determine the necessary null distributions.

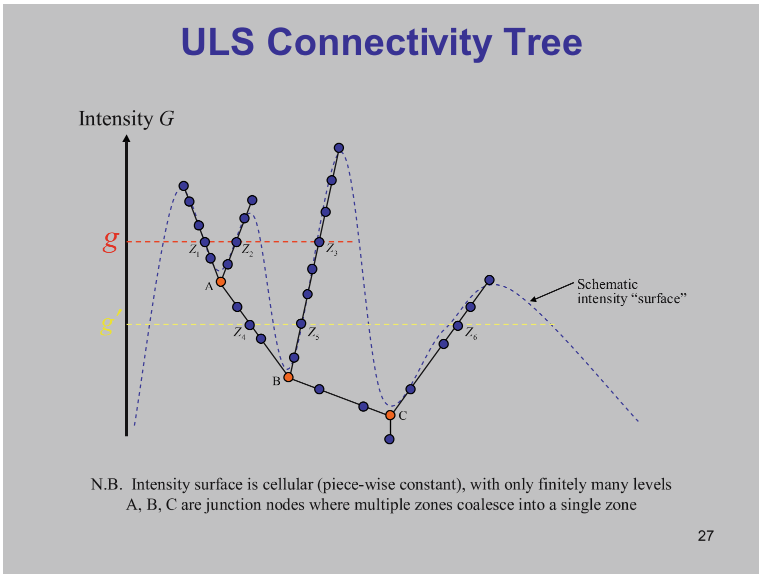
Explication of a likelihood function requires a distributional model (response distribution) for the response Y_a in cell a. This distribution can vary from cell to cell but in a manner that is regulated by the size variable A_a. Thus, A_a enters into the parametric structure of the response distribution. In disease surveillance, response distributions are generally taken as either binomial or Poisson, leading to comparatively simple likelihood functions.

Limitations of Current Scan Statistic Methodology Available scan statistic software suffers from several limitations. First, circles have been used for the scanning window, resulting in low power for detection of irregularly shaped clusters (Fig. 3). Second, the response variable has been defined on the cells of a tessellated geographic region, preventing application to responses defined on a network (stream network, water distribution system, highway system, etc.). Third, response distributions have been taken as discrete (specifically, binomial or Poisson). Finally, the traditional scan statistic returns only a point estimate for the hotspot but does not attempt to assess estimation uncertainty. All of these limitations are addressed by the innovation proposed next.

The Proposed Approach In the approach to the scan statistic, the geometric structure that carries the numerical information is an abstract graph (Fig. 4) consisting of (1) a finite collection of vertexes and (2) a finite set of edges that join certain pairs of distinct vertexes. A tessellation determines such a graph: vertexes are the cells of the tessellation and a pair of vertexes is joined by an edge whenever the corresponding cells are adjacent. A network determines such a graph directly. Each vertex in the graph carries three items of information: (1) a size variable that is treated as known and nonrandom, (2) a response variable whose value is regarded as a realization of some



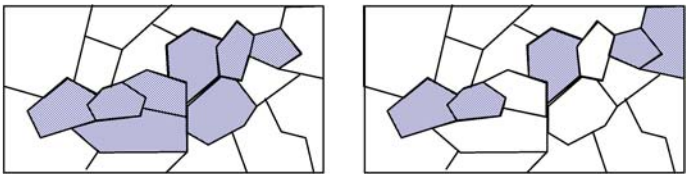
Hotspot Detection, Prioritization, and Security, Figure 3 Scan statistic zonation for circles (**a**) and space–time cylinders (**b**)



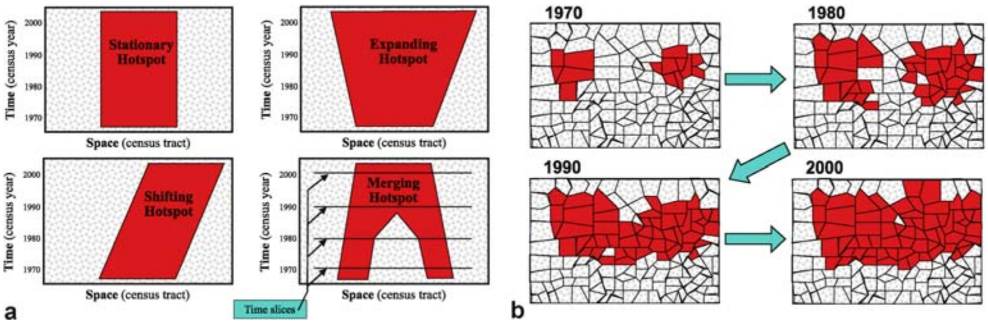
Hotspot Detection, Prioritization, and Security, Figure 4 Upper level set connectivity tree

probability distribution, and (3) the probability distribution itself, which is called the response distribution. Parameters of the response distribution may vary from vertex to vertex, but the mean response (i. e., expected value of the response distribution) should be proportional to the value of the size variable for that vertex. The response rate is the ratio response/size and a hotspot is a collection of vertexes for which the overall response rate is unusually large.

ULS Scan Statistic A new version of the spatial scan statistic is designed for detection of hotspots of arbitrary shapes and for data defined either on a tessellation or a network. This version looks for hotspots among all connected components (Fig. 5) of ULSs of the response rate and is therefore called the ULS scan statistic [1,2]. The method is adaptive with respect to hotspot shape since candidate hotspots have their shapes determined by the data rather than by some a priori prescription like circles or ellipses. This data dependence will be taken into account in the Monte Carlo simulations used to determine null distributions for hypothesis testing. It will also compare performance of the ULS scanning tool with that of the traditional spatial scan statistic. The key element here is enumeration of a searchable list of candidate zones Z. A zone is, first of all, a collection of vertexes from the abstract graph. Secondly, those vertexes should be connected (because a geographically scattered collection of vertexes would not be a reasonable candidate for a "hotspot." Even with this connectedness limitation, the number of candidate zones is too large for a maximum likelihood search in all but the smallest of graphs. The list of zones is reduced to a searchable size in the following way. The response rate at vertex a is $G_a = Y_a / A_a$. These rates determine a function $a \rightarrow G_a$ defined over the vertexes in the graph. This function has only finitely many values (called levels) and each level g determines an ULS of U_g defined by $\{a : G_a > g\}$. ULSs do not have to be connected but each ULS can be decomposed into the disjoint union of connected components. The list of candidate zones Z for the ULS scan statistic consists of all connected components of all ULSs. This list of candidate zones is denoted by Ω_{ULS}. The zones in Ω_{ULS} are certainly plausible as potential hotspots since they are portions of ULSs. Their number is small enough for practical maximum likelihood search; in fact, the size of Ω_{ULS} does not exceed the number of vertexes in the abstract graph (e. g., the number of cells in the tessellation). Finally, Ω_{ULS} becomes a tree under set inclusion, thus facilitating computer representation. This tree is called the ULS-tree; its nodes are the zones $Z \in \Omega_{\text{ULS}}$ and are therefore collections of vertexes from the abstract graph. Leaf nodes are



Hotspot Detection, Prioritization, and Security, Figure 5 Connectivity for tessellated regions. The collection of *shaded cells* in **a** is connected and, therefore, constitutes a zone. The collection in **b** is not connected



Hotspot Detection, Prioritization, and Security, Figure 6 The four diagrams in **a** depict different types of space-time hotspots. The spatial dimension is shown schematically on the horizontal and time is on the vertical. The four diagrams in **b** show the trajectory (sequence of time slices) of a merging hotspot

(typically) singleton vertexes at which the response rate is a local maximum; the root node consists of all vertexes in the abstract graph.

Finding the connected components for an ULS is essentially the issue of determining the transitive closure of the adjacency relation defined by the edges of the graph. Several generic algorithms are available in the computer science literature.

It is important to note that the above approach provides a mathematical and computational structure in the form of a ULS tree for zonal parameter space. And it can help cast and recast the methodological problems and approaches in terms of the quality and quantity of the tree and its nodality [2].

Typology of Space–Time Hotspots

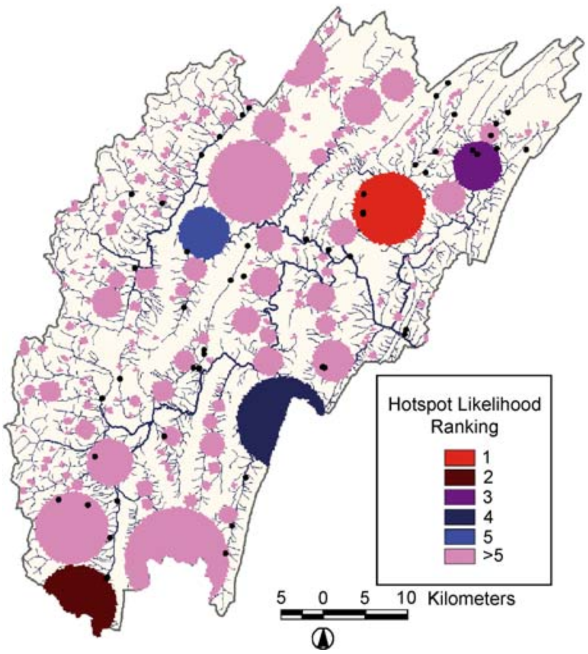
Scan statistic methods extend readily to the detection of hotspots in space–time. The space–time version of the circle-based scan employs cylindrical extensions of spatial circles and cannot detect the temporal evolution of a hotspot. The space–time generalization of the ULS scan detects arbitrarily shaped hotspots in space–time. This helps classify space–time hotspots into various evolutionary types, a few of which appear in Fig. 6a. The merging hotspot is particularly interesting because, while it comprises a connected zone in space–time, several of its time slices are spatially disconnected.

Hotspot Prioritization with Multiple Stakeholders

At times, several hotspots are discovered, and in response to several stakeholders, resulting in their criteria, scores, and/or indicators; the hotspots need to be prioritized and ranked without crunching the indicators into an index. This gives rise to a data matrix of rows for hotspots and columns for indicator scores. The problem becomes a partial order theory problem and has been addressed in [3]. The paper gives many references to other relevant literature. Broadly speaking, this paper is concerned with the question of ranking a finite collection of objects when a suite of indicator values is available for each member of the collection. The objects can be represented as a cloud of points in indicator space, but the different indicators (coordinate axes) typically convey different comparative messages and there is no unique way to rank the objects while taking all indicators into account. A conventional solution is to assign a composite numerical score to each object by combining the indicator information in some fashion. Consciously or otherwise, every such composite involves judgments (often arbitrary or controversial) about tradeoffs or substitutability among indicators.

Rather than trying to combine indicators, it is viewed that the relative positions in indicator space determine only a partial ordering and that a given pair of objects may not be inherently comparable. Working with Hasse diagrams of the partial order, the collection of all rankings is studied that are compatible with the partial order (linear extensions). In this way, an interval of possible ranks is assigned to each object. The intervals can be very wide, however. Noting that ranks near the ends of each interval are usually infrequent under linear extensions, a probability distribution is obtained over the interval of possible ranks. This distribution, called the rank-frequency distribution, turns out to be unimodal (in fact, log-concave) and represents the degree of ambiguity involved in attempting to assign a rank to the corresponding object.

Stochastic ordering of probability distributions imposes a partial order on the collection of rank-frequency distributions. This collection of distributions is in one-to-one correspondence with the original collection of objects and the induced ordering on these objects is called the cumulative rank-frequency (CRF) ordering; it extends the original partial order. Although the CRF ordering need not be linear, it can be iterated to yield a fixed point of the CRF



Hotspot Detection, Prioritization, and Security, Figure 7 Freshwater stream network, water pollution, and management scale

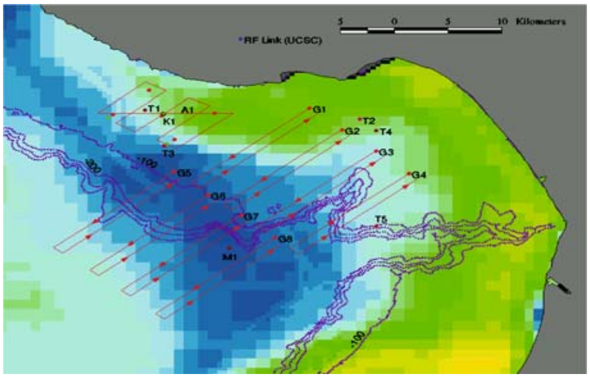
operator. It is hypothesized that the fixed points of the CRF operator are exactly the linear orderings. The CRF operator treats each linear extension as an equal "voter" in determining the CRF ranking. It is possible to generalize to a weighted CRF operator by giving linear extensions differential weights either on mathematical grounds (e. g., number of jumps) or empirical grounds (e. g., indicator concordance). Explicit enumeration of all possible linear extensions is computationally impractical unless the number of objects is quite small. In such cases, the rank-frequencies can be estimated using discrete MCMC methods. See [3] for details.

Key Applications

Broadly speaking, the proposed geosurveillance project and its forum identify several case studies around the world, a selection of illustrative applications and case studies is provided here.

Ecological Indicators and Early Warning

Most states in the mid-Atlantic are in the process of assessing the condition of their aquatic resources. However, their efforts encounter three major stumbling blocks to effective identification of impaired areas and their restoration. First, one impediment has been the lack of reliable ecological indicators that are effective at relevant management scales.



Hotspot Detection, Prioritization, and Security, Figure 8 Oceanic surveillance mobile sensor network

Secondy, to interpret the meaning of any set of indicators, it is necessary to compare the results of monitoring to a relevant and sustainable standard or benchmark, i. e., the best attainable condition for the region or landscape type (e. g., an agricultural versus a forested watershed). Lastly, a method of prioritization is required for cost-effective and timely management. Geoinformatic techniques can address all three obstacles.

The case study (Fig. 7) utilizes the SatScan statistic to prioritize portions of the Upper Juniata watershed for restoration efforts, utilizing the presence of agricultural activities on high slope areas as a predictor of degraded aquatic resource condition. The hotspot areas also provide candidate sites for restoration activities, since management options for this type of land use are already available. Use of the SatScan statistic can therefore provide both a condition assessment, as well as an identification of sites for restoration targeting. While the SatScan statistic is shown to be quite useful in environmental management decisions, the implications of analytical techniques, such as hotspot size, must be taken into consideration [8].

Tasking of a Self-Organizing Oceanic Surveillance Mobile Sensor Network The Autonomous Ocean Sampling Network Simulator (AOSN) is used to study coordination and control strategies for high-resolution, spatiotemporally coordinated surveys of oceanographic fields such as bathymetry, temperature, and currents using autonomous unmanned undersea vehicles. Currently, the network of mobile sensor platforms (Fig. 8) is autonomous and self-organizing, once given high-level tasking from an external tactical coordinator. This case study proposes to use ULS scan statistic theory to identify hotspots in data gathered by the sensor network and use this information to dynamically task mobile sensor platforms so that more data can be gathered in the areas of interest. By detecting hotspots and tasking accordingly, network resources are not wasted on mapping areas of little change. The ability of the sensor network to dynamically task its own components expands the network's mission and increases the reactivity to changing conditions in a highly dynamic environment [9,10,11].

Cyber Security and Computer Network Diagnostics

Securing the nation's computer networks from cyber attacks is an important aspect of national Homeland Security. Network diagnostic tools aim at detecting security attacks on computer networks. Besides cyber security, these tools can also be used to diagnose other anomalies such as infrastructure failures, and operational aberrations. Hotspot detection forms an important and integral part of these diagnostic tools for discovering correlated anomalies. It is constructive to develop a network diagnostic tool at a functional level. The goal of network state models is to obtain the temporal characteristics of network elements such as routers, typically in terms of their physical connectivity, resource availability, occupancy distribution, blocking probability, etc. There is some prior work [12] in developing network state models for connectivity, and resource availability. Models have been also developed for studying the equilibrium behavior of multidimensional loss systems. The probabilistic finite state automaton (PFSA) describing a network element can be obtained from the output of these state models. A time-dependent crisis-index is determined for each network element, which measures their normal behavior pattern compared to crisis behavior. The crisis-index is the numerical distance between the stochastic languages generated by the normal and crisis automata. Use of the variational distance between probability measures seems attractive, although other distances can also be considered. The crisis behavior can be obtained from past experience. The crisis indices over a collection of network elements are then used for hotspot detection using scan statistic methodology. These hot spots help to detect coordinated security attacks geographically spread over a network.

West Nile Virus: An Illustration of the Early Warning Capability of the Scan Statistic

Since the 1999 West Nile (WN) virus outbreak in New York City, health officials have been searching for an early warning system that could signal increased risk of human WN infection, and provide a basis for targeted public education and increased mosquito control. Birds and mosquitoes with laboratory evidence of WN virus preceded most human infections in 2000, but sample collection and laboratory testing are time-consuming and costly. The cylinder-based space–time scan statistic for detecting small area clustering of dead bird reports have been assessed for its utility in providing an early warning of WN virus activity [13].

Hotspot Detection, Prioritization, and Security, Figure 9 West Nile virus in New York City

All unique nonpigeon dead bird reports were categorized as "cases" if occurring in the prior 7 days and "controls" if occurring during a historic baseline. The most likely cluster area was determined using the scan statistic and its statistical significance was evaluated using Monte Carlo hypothesis testing. Analyses were performed in a prospective simulation for 2000 and in real-time during 2001 (Fig. 9).

For the 2000 data, dead bird clustering was found in Staten Island over 4 weeks prior to laboratory evidence of WN virus in birds and mosquitoes from this area. Real-time implementation in 2001 led to intensified larval control in eastern Queens, over 3 weeks prior to laboratory confirmation from this cluster area. Dead bird clusters were identified around the residences of five of seven human infections, from 0 to 40 days (median 12) prior to the onset of illness, and 12–45 days (median 17) prior to human diagnosis.

It has been concluded that scan statistical cluster analysis of dead bird reports may provide early warning of viral activity among birds and of subsequent human infections. Analysis with the ULS space–time scan statistic will be worthwhile. Since the latter allows for arbitrarily shaped clusters in both the spatial and temporal dimensions, there is potential for earlier detection with a reduced false alarm rate.

Wireless Sensor Networks for Object Recognition and Tracking in Geotelemetry

Technology has seen rapid developments in geographic information systems (GIS), wireless communication, global positioning system (GPS) components, and miniature sensor detection devices. It is crucial to integrate these advances with a GIS framework and accomplish a miniature, low-cost wireless communication and GPS capability (geotelemetry) that transmits data, including event, time, and location from a mesh network of mixed miniature sensors.

Remotely deployed, self-sufficient, and miniature smart sensor devices are considered essential in geospatial surveillance. Such sensors extend situational awareness in high-risk or denied areas by providing early warning or continuing assessment of the situation with dynamic spatial mapping.

For activities that tend to take place over wide geographical areas, such as battlefields, agriculture areas, mining districts, oil fields, refineries, and resource management areas, and for use in environmental monitoring and security, randomly distributed remote smart sensors can monitor critical conditions and events.

The data-handling architecture for the wireless devices of the sensor networks need to incorporate their extreme resource constraints—energy, storage, and processing—and spatiotemporal interpretation of the physical world in the design, cost model, and metrics of evaluation. For this important emerging problem area, investigations are in the beginning stages for appropriate mathematical, statistical, and computational tools, such as decentralized Kalman filters and the communication issues, decentralized data fusion and control, distributed target classification and tracking, support vector machines for vehicle classification, distributed detection and decision fusion, and appropriate incorporation of GPS, GIS, and related information extraction. It is only timely to evaluate and advance appropriate mathematical and statistical tools for robust and flexible multisensor distributed environments.

Object identification and tracking may require data analysis from single or multiple sensors. For example, identification of temperature can be made by a single temperature sensor, but identification of battle tanks and other vehicles can utilize data from multiple sensors, such as vibration and sound. The data from such sensors must be fused before statistical analyses for detection, classification, and prediction can be done and before a confidence level can be determined. Such analysis, including anticipated signatures of detected objects, requires the use of advanced methods, like the Kalman filter and Bayesian theory. Future prediction of object signature is essential to extrapolate abnormal behavior and send a Warning [14]. Figure 10 may be helpful in capturing the relevant scenario in geotelemetry utilizing unmanned aerial vehicles (UAVs) for purposes of object recognition and tracking in the form of the temporal dynamics of detected geospatial hotspots using spatiotemporal hotspot detection tools of ULS systems.

Hotspot Detection, Prioritization, and Security, Figure 10 Geotelemetry enabled sensor nodes deployed by a UAV into a wireless ad hoc mesh network: Transmitting data and coordinates to TASS (Term Agreement for Survey Services) and geographic information systems (GIS) support systems

Future Directions

This article briefly describes a prototype geoinformatic hotspot surveillance system for hotspot delineation and prioritization and provides a variety of case studies and illustrative applications of societal importance. The prototype system consists of modules for hotspot detection and delineation, and for hotspot prioritization. It will be productive and opportune to maintain and strengthen this momentum.

Surveillance geoinformatics of hotspot detection and prioritization is a critical need of the twenty-first century. Next generation decision support system within this context is crucial. It will be productive to build on the present effort in the directions of prototype and user-friendly methods, tools, and software, and also in the directions of thematic groups, working groups, and case studies important at various scales and levels. The authors have such a continuation effort in progress within the context of digital governance with NSF support and would welcome interested readers to join in this collaborative initiative.

Acknowledgements

This material is based upon work supported by (a) the National Science Foundation under Grant No. 0307010, and (b) the United States Environmental Protection Agency under Grants No. CR-83059301 and No. R-828684-01. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the agencies.

Recommended Reading

1. Patil, G.P., Taillie, C.: Upper level set scan statistic for detecting arbitrarily shaped hotspots. Environ. Ecol. Stat. **11**, 183–197 (2004)
2. Patil, G.P., Duczmal, L., Haran, M., Patankar, P.: On PULSE: the progressive upper level set scan statistic system for geospatial and spatiotemporal hotspot detection. In: The 7th Annual International Conference on Digital Government Research, San Diego, CA, May 21–24, 2006
3. Patil, G.P., Taillie, C.: Multiple indicators, partially ordered sets, and linear extensions: multi-criterion ranking and prioritization. Environ. Ecol. Stat. **11**, 199–228 (2004)
4. Patil, G.P., Bhat, K.S., Kase, M.: Multicriterion prioritization with differential weights, stepwise aggregation, Hasse Diagrams, and Poset cumulative rank frequency operators. In: The 7th Annual International Conference on Digital Government Research, San Diego, CA, May 21–24, 2006
5. Kulldorff, M.: A spatial scan statistic. Commun. Stat Theor. Methods **26**, 1481–1496 (1997)
6. Kulldorff, M., Feuer, E., Miller, B., Freedman, L.: Breast cancer in northeast United States: a geographic analysis. Am. J. Epidemiol. **146**, 161–170 (1997)
7. Waller, L.: Methods for detecting disease clustering in time or space. In: Brookmeyer, R., Stroup, D. (eds.) Monitoring the Health of Populations: Statistical Methods and Principles in Public Health Surveillance, pp. 167–202. Oxford University Press, New York (2004)
8. Wardrop, D.H.: Digital tools for development, testing, and application of ecological indicators and early warning. In: Workshop on Digital Governance and Hotspot GeoInformatics at the Digital Government Research Conference, San Diego, CA, 21–24 May 2006
9. Phoha, S., Peluso, E., Stadter, P., Stover, J., Gibson, R.: A mobile distributed network of autonomous undersea vehicles. In: Proceedings of the 24th Annual Symposium and Exhibition of the Association of Unmanned Vehicle Systems International, Baltimore, MD, 3–6 June 1997
10. Phoha, S., Eberbach, E., Brooks, R.: Coordination of multiple heterogeneous autonomous undersea vehicles (AUVs) Special Heterogeneous Multi-Robot Systems Issue of Autonomous Robots, October (1999)
11. Phoha, S., Peluso, E., Culver, R.L.: A high fidelity ocean sampling mobile network (SAMON) simulator tested for evaluating intelligent control of unmanned underwater vehicles. IEEE J. Ocean Eng. Special Edn, June (2001)
12. Ghosh, D. Acharya, R.: A probabilistic scheme for hierarchical routing. In: Proceedings of Ninth IEEE International Conference on Networks, Bangkok, Thailand, 10–12 2001
13. Mostashari, F., Kulldorff, M., Hartman, J., Miller, J., Kulasekera, V.: Dead bird clusters as an early warning system for West Nile Virus activity. Emerging Infect. Dis. **9**, 641–646 (2003)
14. Ristic, B., Arulampalm, S., Gordon, N.: Beyond the Kalman Filter: Particle Filters for Tracking Applications. Artech House, Boston (2004)

Hurricane Wind Fields, Multivariate Modeling

Kristen M. Foley, Montserrat Fuentes
Department of Statistics, North Carolina State University, Raleigh, NC, USA

Synonyms

Cross-covariance models; Nonseparable multivariate models; Bayesian inference; Statistical space-time modeling; Gaussian; Stationarity; Separability; Matrix, inverse; Determinant; Dimension reduction; Parametric model; MCMC; GIbb's sampling; Markov random field (MRF)

Definition

Multivariate Spatial and Spatiotemporal Processes

Statistical space–time modeling has proven to be an essential tool in the environmental sciences to describe complex spatial and temporal behavior of physical processes. Statistical models also allow for prediction of underlying spatial-temporal processes at new locations and times based on noisy observations. In many cases the data being analyzed consist of multivariate observations, meaning multiple variables have been measured at the same location and time. For example, air monitoring networks often record multiple pollutant levels at a single monitoring site. The same is true for meteorological stations that may record air temperature, humidity, precipitation, etc. When analyzing such data sets, where the variables of interest are highly correlated, modeling the physical processes jointly will improve estimation and prediction results, compared to treating each variable as an independent process.

Cross-Covariance Function In statistical space–time modeling of Gaussian processes, the covariance function is used to describe the spatial and temporal dependency structure of the observations. In the multiple variable setting, the specification of a cross–covariance function allows for additional dependence between observations of different variables. In general, let $\mathbf{Z}(\mathbf{s}, t)$ be a $p \times 1$ vector valued spatial–temporal process. For r locations $\mathbf{s}_1, \ldots, \mathbf{s}_r \in \mathcal{R}^d$ (d-dimensional Euclidean space) and m observations over time $t_1, \ldots, t_m \in \mathcal{R}$ there are $n = r \cdot m$ observations for each variable and np total observations. Let the $(np \times 1)$ data vector be defined such that $\mathbf{Z} = (\mathbf{Z}_1^{\mathrm{T}}, \ldots, \mathbf{Z}_p^{\mathrm{T}})^{\mathrm{T}}$ where $\mathbf{Z}_j = (Z_j(\mathbf{s}_1, t_1),\ Z_j(\mathbf{s}_1, t_2), \ldots, Z_j(\mathbf{s}_1, t_m),\ Z_j(\mathbf{s}_2, t_1), \ldots, Z_j(\mathbf{s}_r, t_m))^{\mathrm{T}}$. For example, for measurements of wind speed and direction from coastal buoy stations, $p = 2$, each location $\mathbf{s}$ is defined by degrees longitude and latitude and $t_1,\ t_2, \ldots, t_m$, may be hourly time increments measured in Coordinated Universal Time (UTC). For convenience of notation, let $\mathbf{x} = (\mathbf{s}, t) \in \mathcal{R}^{d+1}$ so that $\mathbf{Z}(\mathbf{s}_k, t_l) = \mathbf{Z}(\mathbf{x}_{kl})$. Thus for statistical modeling of multivariate processes, rather than treating the p variables of interest as spatially and temporally independent and modeling each separately, the cross-covariance function, $\text{Cov}(Z_j(\mathbf{s}_k, t_l), Z_{j'}(\mathbf{s}_{k'}, t_{l'}))$, defines how two variables covary over space and time. For further details on statistical modeling of multivariate space–time data see [7,12,18] and references therein.

Stationarity and Separability Many classes of univariate space–time models rely on the assumption that the processes of interest are stationary, meaning that the structure of the spatial–temporal covariance does not change with location or time. More precisely, $Z_j(\mathbf{x})$ is a *weakly stationary* spatial–temporal process if it has a constant mean function $\mu(\mathbf{x}) = \mu$, the variance of $Z_j(\mathbf{x})$ exists for all $\mathbf{x}$, and the covariance of $Z_j(\mathbf{x}_{kl})$ and $Z_j(\mathbf{x}_{k'l'})$ is a function of the separation vector between $\mathbf{x}_{kl}$ and $\mathbf{x}_{k'l'}$ ($\mathbf{h} = \mathbf{x}_{kl} - \mathbf{x}_{k'l'}$). For example, for $m = 1$, a standard geostatistical model for spatial covariance is the exponential function, $\text{Cov}(Z_j(\mathbf{x}_{kl}), Z_j(\mathbf{x}_{k'l'})) = \sigma^2 \exp(\mathbf{h}/d)$ where σ^2 represents the total variance of the process, know as the *sill* parameter, and *3*d* is the *effective range* parameter that represents the distance at which two measurements become practically uncorrelated. Under this model the correlation between observations at two locations will decrease exponentially as the distance between these sites increases. Hence there are only two parameters that must be estimated in order to define the full $n \times n$ covariance matrix for variable Z_j.

The simplest cross–covariance model is separable in space and time, i. e. there is no interaction in the spatial and temporal variability, and also separable in the sense that the correlation structure between different variables does not depend on spatial scale or temporal scale. Thus the $np \times np$ cross–covariance matrix for $\mathbf{Z}$, $(\Sigma = \text{Cov}(\mathbf{Z}, \mathbf{Z}))$, can be written as a Kronecker product of the $p \times p$ variance/covariance matrix of the p variables, the $r \times r$ spatial covariance matrix and the $m \times m$ temporal covariance matrix.

$$\Sigma = \Sigma_Z \otimes \Sigma_S \otimes \Sigma_T \,. \qquad (1)$$

Again, a separable model will typically have fewer parameters than a nonseparable specification. For example,

$$\text{Cov}(\mathbf{Z}(\mathbf{s}_k, t_l), \mathbf{Z}(\mathbf{s}_{k'}, t_{l'})) = \rho_S(\mathbf{s}_k - \mathbf{s}_{k'}; \theta_S)\rho_T(t_l - t_{l'}; \theta_T)\mathbf{T} \qquad (2)$$

where $\rho_S(.)$ is a valid univariate spatial correlation function with parameters θ_S, $\rho_T(.)$ is a one-dimensional temporal correlation function with parameters θ_T and $\mathbf{T}$ a $p \times p$ positive definite variance-covariance matrix for the vector of variables, $\mathbf{Z}$, at a fixed location and time.

Prediction at a new location and time for Gaussian processes, based on all available data, requires the calculation of the inverse and the determinant of the $np \times np$ matrix, **\Sigma**. A model such as (1) will greatly decrease computational cost of these calculations especially for large n. However, this model is limited by the fact that it does not allow for spatial–temporal interactions, or for the correlation between variables to evolve over space or time.

Although the assumptions of stationarity and separability may simplify model estimation, they are not always realistic, especially when working with data sets that cover large geographic domains or are observed over long periods of time. For example, Fuentes et al. describe how the spatial structure of ozone concentrations varies across the Eastern United States due to differences in geography and proximity to pollutant sources [4]. If the nonstationarity is not accounted for in these cases, then predictions and the prediction variance will not be reliable, since predicted values may be unduly influenced by observations with a very different spatial structure. For a review of various classes of statistical space–time models developed to address issues of stationarity and separability for univariate processes see [1].

H

Linear Model of Coregionalization

The linear model of coregionalization (LMC) approach was first introduced as a method of dimension reduction in order to model a set of p multivariate observations as a function of k unobservable factors with $k < p$. For further details see [17]. The LMC approach is also used to construct valid cross-covariance functions by modeling a set of dependent spatial processes as a linear combination of independent spatial processes, e. g. [1,7,15].

For applications in modeling multivariate spatial–temporal processes, the basis of the linear model of coregionalization is to decompose each of p variables into p uncorrelated components. Let $w_j(\mathbf{s}, t)$ be p latent independent Gaussian spatial–temporal processes with mean zero and covariance functions ρ_j, $j = 1, \ldots, p$. The parametrization of these space–time covariance functions will depend on the application. For example, to model a space–time interaction, ρ_j may be nonseparable parametric models based on a 3D Matérn covariance [5]. A multivariate spatial–temporal process is then modeled as:

$$\mathbf{Z}(\mathbf{s}, t) = \mathbf{A}\mathbf{w}(\mathbf{s}, t) \,, \qquad (3)$$

where $\mathbf{w}(\mathbf{s}, t) = (w_1(\mathbf{s}, t), \ldots, w_p(\mathbf{s}, t))^{\mathrm{T}}$ and $\mathbf{A}$ is a $p \times p$ weight matrix that determines the covariance between the

p variables. Without loss of generality, $\mathbf{A}$ is defined as a full rank lower triangular matrix. This implies $E(\mathbf{Z}(\mathbf{s}, t)) = \mathbf{0}$ and

$$\mathrm{Cov}(\mathbf{Z}(\mathbf{s}_k, t_l), \mathbf{Z}(\mathbf{s}_{k'}, t_{l'})) = \sum_{j=1}^{p} \rho_j((\mathbf{s}_k, t_l), (\mathbf{s}_{k'}, t_{l'}); \theta_j)\mathbf{T}_j\,, \tag{4}$$

where $\mathbf{T}_j = \mathbf{a}_j\mathbf{a}_j^{\mathrm{T}}$ and $\mathbf{a}_j$ is the j^{th} column of $\mathbf{A}$. The cross–covariance function is nonseparable, meaning the between-site variability cannot be separated from the variability between measures taken at the same site,

$$\Sigma = \sum_{j=1}^{p} \mathbf{T}_j \otimes \mathbf{R}_j\,, \tag{5}$$

for $\mathbf{R}_j$ a set of p $n \times n$ matrices with elements defined by $\rho_j((\mathbf{s}_k, t_l), (\mathbf{s}_{k'}, t_{l'}); \theta_j)$, $j = 1, \ldots, p$. Additional properties of the cross–covariance depend on the form of the space–time functions ρ_j. Thus for a nonstationary model for ρ_j with parameters θ_j, the cross–covariance will also be nonstationary. Alternatively, for ρ_j stationary correlation functions, $\sum_{j=1}^{p} \mathbf{T}_j = \mathbf{T}$ is the covariance matrix for $\mathbf{Z}(\mathbf{s}_k, t_l)$. For example for $p = 2$ and $(\mathbf{A})_{ij} = a_{ij}$:

$$\mathbf{T} = \begin{pmatrix} a_{11}^2 & a_{11}a_{21} \\ a_{11}a_{21} & a_{21}^2 + a_{22}^2 \end{pmatrix} \tag{6}$$

Note that a separable covariance model is just a special case of the LMC model where $\rho_j = \rho$, a stationary correlation function, for all $j = 1, \ldots, p$. In this case, $\mathrm{Cov}(\mathbf{Z}, \mathbf{Z}) = \mathbf{T} \otimes \mathbf{R}$. This is referred to as an *intrinsic* multivariate correlation model, because the correlation structure of the set of variables is independent of the spatial and temporal correlation [17].

Holland Model for Cyclostrophic Wind Balance

The Holland model describes a deterministic model for the region of hurricane surface winds where the Coriolis force is small relative to the pressure gradient and centrifugal forces [9]. This region of surface winds is said to be in cyclostrophic balance. The Holland formula for surface wind speed at location $\mathbf{s}$, time t is:

$$W(\mathbf{s}, t) = \left[\frac{B}{\rho} \left(\frac{Rmax}{r(\mathbf{s}, t)} \right)^B (Pn - Pc_t) \exp^{-(Rmax/r(\mathbf{s},t))^B} + \left(\frac{r(\mathbf{s}, t)f}{2} \right)^2 \right]^{1/2} - \frac{r(\mathbf{s}, t)f}{2}\,. \tag{7}$$

f is the Coriolis parameter, but for low latitude hurricanes the terms associated with f are neglected in (7). Pn is the ambient pressure, Pc_t is the hurricane central pressure at time t, ρ is the air density (fixed at 1.2 kg m^{-3}), and $r(\mathbf{s}, t)$ is the distance from the hurricane center (at time t) to location $\mathbf{s}$.

The Holland formula for wind speed is used to create an axissymmetric wind field, meaning the wind speeds at different azimuthal directions are assumed to be the same. Specifically, the wind vector may be decomposed into u (East-West) and v (North-South) orthogonal components:

$$u^H(\mathbf{s}, t) = W(\mathbf{s}, t)\sin(\phi(\mathbf{s}, t)) \tag{8}$$

$$v^H(\mathbf{s}, t) = W(\mathbf{s}, t)\cos(\phi(\mathbf{s}, t)) \tag{9}$$

where $\phi(\mathbf{s}, t)$ is the inflow angle at site $\mathbf{s}$ across circular isobars toward the storm center (at time t). Parameter B represents the shape and size of the hurricane vortex and determines the maximum possible wind speed in the storm based on the observed central pressure (lower pressure induces larger wind speeds). The parameter $Rmax$ is the radius of maximum sustained wind of the hurricane. Recent adaptations to this formulation may be found in [20].

Historical Background

An important issue of modeling spatial–temporal environmental processes is how to effectively utilize different sources of available information. Over the past decade there has been an increase in the amount of available real-time observations of geophysical and biological processes. Whether the variables of interest are directly observed (e. g. data from buoys or anemometers) or indirectly observed (e. g. satellite data), these observations may be at very different scales in space and time. Different sources of observational data also differ in the type and magnitude of systematic error and random variability that can be expected. In addition, there may be substantial scientific knowledge about the physical processes being measured that is described in the form of deterministic equations. Such models, based on the dynamics and mechanics of atmospheric, oceanic or biological processes, typically provide information at higher temporal and spatial resolution than data from observational networks. Errors and biases in these deterministic models are still inevitable, due to simplified or neglected physical processes, or mathematical approximations used in the physical parametrization.

Past research has demonstrated the use of a Bayesian framework for combining information from disparate data sources with scientific knowledge of physical processes. Fuentes and Raftery offer a cohesive approach to combine observational data and numerical model output, while accounting for bias and measurement error in these observations [6]. Foley and Fuentes extend this framework to allow for the estimation of unknown parameters in physically-based deterministic models [3]. Alternatively, Wikle et al. present a spatial–temporal model that combines two very different datasets for surface wind fields over the tropical oceans [19]. Through a series of conditional (hierarchical) models, Wikle et al. specify a statistical data model and process model that is used to approximate known physical dynamics.

Here the Bayesian methodology from these past studies is extended to describe a statistical model for multivariate datasets. This approach accounts for issues of stationarity and separability in modeling multivariate spatial and temporal processes as well as provides a framework for combining data from multiple sources and incorporating physical information through deterministic models. Uncertainty in the parametrization of the deterministic equations and bias in the observed data are implicitly accounted for. This framework is then applied for coastal ocean prediction along the Eastern coastline utilizing the operational methods and observational data currently available to the National Hurricane Center (NHC) as a case study.

Scientific Fundamentals

Some common features of applications in the physical and biological sciences are that information on the processes of interest exist in the form of observational data and physical or biological parametrizations. A hierarchical Bayesian modeling framework allows for estimation of the parameters of the multivariate statistical model, as well as parameters of physically-based deterministic models while accounting for potential additive and multiplicative bias in the observed data sets. Figure 1 depicts how the Bayesian modeling framework is used to combine different data sets. The modeling of the multivariate spatial–temporal processes may be described in stages. These stages are used in the subsequent estimation and prediction algorithm.

Stage 1: In Stage 1 different sources of data are represented in terms of an underlying true (unobserved) multivariate process $\mathbf{V}(\mathbf{s}, t)$. Let $\hat{\mathbf{V}}^a_{(n_a p\times 1)}$ and $\hat{\mathbf{V}}^b_{(n_b p\times 1)}$ represent data vectors of n_a and n_b observations, respectively, from two disparate sources of information on the same p variables of interest. The sources of data may have very different spatial and temporal resolutions. For example this may be data from two different air quality monitoring networks that use different methods for measuring pollutant concentrations. Or $\hat{\mathbf{V}}^a$ may be satellite data such as measured cloud cover or surface wind speeds and $\hat{\mathbf{V}}^b$ may be surface data that is directly observed by weather stations or buoys.

The data model includes measurement error terms, $\epsilon^b(\mathbf{s}, t)$ and $\epsilon^a(\mathbf{s}, t)$, as well as additive and multiplicative bias functions, $\mathbf{c}(\mathbf{s}, t) = (c_1(\mathbf{s}, t), \ldots, c_p(\mathbf{s}, t))^{\mathrm{T}}$ and $\mathbf{m}(\mathbf{s}, t)$ a diagonal matrix with diagonal elements $(m_1(\mathbf{s}, t), \ldots, m_p(\mathbf{s}, t))$. Bias will likely exist, to some extent, in both sources of data, but it is not possible to accurately estimate the bias of both data sources without external information from historical analyses or validation studies. If this prior information is available it may also be incorporated by fixing some of the bias parameters. It is often the case, that the magnitude of the bias in one data source, for example $\hat{\mathbf{V}}^a$, is known to be much larger than potential bias in the second data source. A reasonable solution is then to estimate the bias of $\hat{\mathbf{V}}^a(\mathbf{s}, t)$ *with respect to* $\hat{\mathbf{V}}^b(\mathbf{s}, t)$. The statistical model for the data at location $\mathbf{s}$ and time t is:

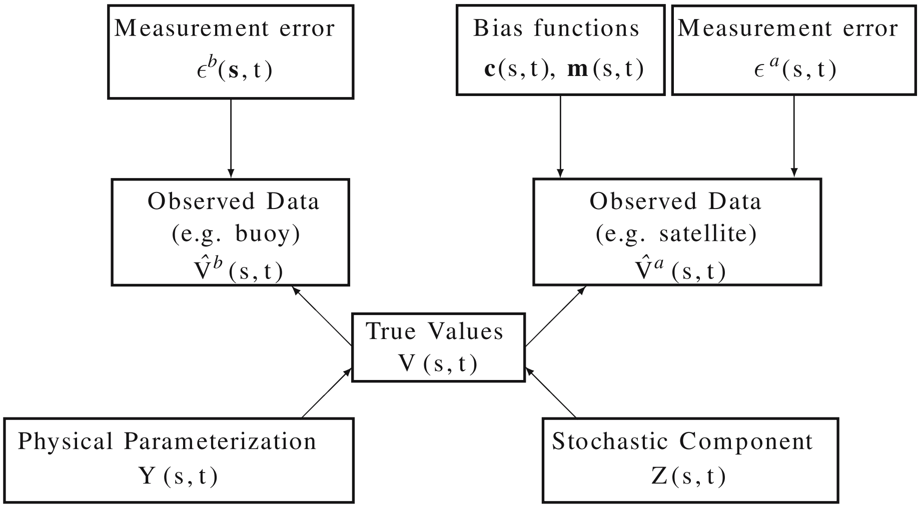
$$\hat{\mathbf{V}}^a(\mathbf{s}, t) = \mathbf{c}(\mathbf{s}, t) + \mathbf{m}(\mathbf{s}, t)\mathbf{V}(\mathbf{s}, t) + \epsilon^a(\mathbf{s}, t) \qquad (10)$$

$$\hat{\mathbf{V}}^b(\mathbf{s}, t) = \mathbf{V}(\mathbf{s}, t) + \epsilon^b(\mathbf{s}, t)\,. \qquad (11)$$

In this way if there is a more temporally or spatially sparse dataset that is a direct measure of the variables of interest, this information is used to adjust more spatially or temporally rich data that is known to have large systematic errors, such as remote sensor data. Each data source has limitations and strengths and accounting for that in a statistical model will improve modeling results.

The additive and multiplicative bias terms may be modeled as smooth functions of spatial location or functions of additional spatial covariates. In general, assume the additive and multiplicative bias terms are functions of the parameters θ^c and θ^m, respectively. The complexity of the bias functions will be limited by the amount of available data in $\hat{\mathbf{V}}^b$. The measurement error processes, ϵ^a and ϵ^b, are treated as Gaussian processes with error covariance matrices, Σ^a and Σ^b. The measurement error terms are assumed independent of one another and the underlying process $\mathbf{V}(\mathbf{s}, t)$. The parametrization of the error covariance matrix will depend on information about how the observations are measured. For example, it may or may not be reasonable to assume that the errors of different variable measurements are independent with the same error variability. In this case $\Sigma^a = \sigma_a^2\mathbf{I}$ and $\Sigma^b = \sigma_b^2\mathbf{I}$. More complex measurement error may be expected, but again the statistical model will be limited by what can be estimated with available data.

Stage 2: Stage 2 is a statistical model for the unobserved "true" process based on deterministic physical equations and a stochastic spatial–temporal residual component. Let $\mathbf{Y}(\mathbf{s}, t)$ represent output from deterministic equations for the p variables at a given location and time. It is assumed



Hurricane Wind Fields, Multivariate Modeling, Figure 1
Model framework for fusing physical models and data

this is a deterministic formula conditioned on fixed parameters, θ^Y and known covariate data. Note that output from numerical models could also be treated as a different source of data in Stage 1 as in [11]. In the framework described here, the physically based models are represented with a stochastic parametrization that accounts for uncertainty in the parameters of the deterministic equations. In addition, a multivariate spatial–temporal process, $\mathbf{Z}(\mathbf{s}, t)$, is used to model the variability of the true process that cannot be captured by any parametrization of physically-based formulation. The model for the underlying process variables at location $\mathbf{s}$ and time t is defined as:

$$\mathbf{V}(\mathbf{s}, t) = \mathbf{Y}(\mathbf{s}, t) + \mathbf{Z}(\mathbf{s}, t)\,. \tag{12}$$

The stochastic component in (12) is modeled based on the Linear Model of Coregionalization (LMC). $\mathbf{Z}(\mathbf{s}, t)$ is a linear combination of p mean–zero latent independent Gaussian spatial–temporal processes $w_j(\mathbf{s}, t)$:

$$\mathbf{Z}(\mathbf{s}, t) = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & 0 \\ & \dots & \dots & \\ a_{p1} & a_{p2} & \dots & a_{pp} \end{pmatrix} \begin{pmatrix} w_1(\mathbf{s}, t) \\ w_2(\mathbf{s}, t) \\ \dots \\ w_p(\mathbf{s}, t) \end{pmatrix}. \tag{13}$$

Each process has covariance function $\mathrm{Cov}(w_j(\mathbf{x}_{kl},), w_j(\mathbf{x}_{k'l'})) = \rho_j(\mathbf{x}_{kl}, \mathbf{x}_{k'l'}; \theta_j)$ with parameters θ_j, $j=1,\dots,p$ for $\mathbf{x}_{kl} = (\mathbf{s}_k, t_l)$. These covariance functions do not need to be stationary or separable. Thus all of the spatial–temporal structure is modeled within the latent processes. The structure of the space–time covariance functions will depend on the application. For example, for a nonstationary spatial process, the ρ_j may be modeled through a series independent space–time processes over a set of discrete subregions of stationarity [5].

Bayesian Approach for Estimation and Prediction

In order to predict the variables at a new time or location, estimates are first needed for the parameters of the statistical model. A Bayesian estimation approach is used to compute the posterior density for $\boldsymbol{\Theta} = (\theta^a, \theta^b, \theta^Y, \theta^Z)$ where θ^a and θ^b are the bias and measurement error parameters for each dataset and $\theta^Z = (\mathbf{A}, \{\theta_j\})$ are the parameters for the LMC model for $j = 1, \dots, p$. For $N = n_a + n_b$, let $\hat{\mathbf{V}} = ((\hat{\mathbf{V}}^a)^{\mathrm{T}}, (\hat{\mathbf{V}}^b)^{\mathrm{T}})^{\mathrm{T}}$ be the $Np \times 1$ vector of all measurements from both sources of observational data. Let $\mathbf{Y}$ and $\mathbf{V}$ denote the output from the physical model, and the vector of true process values, respectively, for the locations and times of $\hat{\mathbf{V}}$. Similarly, let $\mathbf{c}_{(n_a p \times 1)}$ and $\mathbf{m}_{(n_a p \times n_a p)}$ be the values of the bias functions evaluated at the locations and times of the dataset $\hat{\mathbf{V}}^a$. Recall, $\mathbf{Y}$, $\mathbf{c}$, and $\mathbf{m}$ are deterministic functions given the values of $(\theta^c, \theta^m, \theta^Y)$.
The posterior predictive distribution conditioned on all available data is used to predict the true process $\mathbf{V}$ at new location $\mathbf{s}_o$ and time t_o. The posterior predictive distribution is of the form:

$$P\left(\mathbf{V}(\mathbf{s}_o, t_o)|\hat{\mathbf{V}}\right) \propto \int P\left(\mathbf{V}(\mathbf{s}_o, t_o)|\hat{\mathbf{V}}, \boldsymbol{\Theta}\right) \times P\left(\boldsymbol{\Theta}|\hat{\mathbf{V}}\right) \mathrm{d}\boldsymbol{\Theta} \tag{14}$$

The first term within the integral is a multivariate normal distribution:

$$P\left(\mathbf{V}(\mathbf{s}_o, t_o)|\hat{\mathbf{V}}, \boldsymbol{\Theta}\right) \sim N\left(\mathbf{Y}(\mathbf{s}_o, t_o) + \tau^{\mathrm{T}}\Sigma^{-1}[\hat{\mathbf{V}} - \mu], \mathbf{T}_o - \tau^{\mathrm{T}}\Sigma^{-1}\tau\right),$$

where $\tau_{(Np \times p)} = \mathrm{Cov}(\mathbf{V}(\mathbf{s}_o, t_o), \hat{\mathbf{V}})$, $\Sigma_{(Np \times Np)} = \mathrm{Cov}(\hat{\mathbf{V}}, \hat{\mathbf{V}})$ and $\mu = ((\mathbf{c} + \mathbf{m}\mathbf{Y})^{\mathrm{T}}, \mathbf{Y}^{\mathrm{T}})^{\mathrm{T}}$ is the output from the deterministic function, accounting for bias in one of the datasets. $\mathbf{T}_o$ is the $p \times p$ variance-covariance matrix for $\mathbf{V}(\mathbf{s}_o, t_o)$ based on the LMC cross-covariance model.

A Markov Chain Monte Carlo (MCMC) approach to sample from the posterior distribution of the parameters is used to estimate (14). Based on K samples from the posterior distributions for the parameters, $\{\mathbf{\Theta}^{(k)}\}$, $k = 1, \ldots, K$, the full posterior density is estimated with the Rao–Blackwellized estimator:

$$P\left(\mathbf{V}(\mathbf{s}_o, t_o)|\hat{\mathbf{V}}\right) = \frac{1}{K}\sum_{k=1}^{K} P\left(\mathbf{V}(\mathbf{s}_o, t_o)|\hat{\mathbf{V}}, \mathbf{\Theta}^{(k)}\right) . \quad (15)$$

The result is an estimate of the full conditional distribution of the underlying process that accounts for uncertainty in the estimation of the parameters in the statistical and physical model.

To sample from the posterior distribution for $\mathbf{\Theta}$, a multiple–stage Gibbs sampling algorithm for the MCMC breaks the problem into a series of conditional distributions. The Gibbs algorithm proceeds in the following stages.

Stage 1: Conditioned on the true wind process $\mathbf{V}$ (updated in Stage 3), obtain the posterior distribution of the parameters, $\theta^a = (\theta^c, \theta^m, \Sigma^a)$, $\theta^b = (\Sigma^b)$, that explain the bias and uncertainty about the data. The posterior distribution for (θ^a, θ^b) is determined by the prior distributions given for each parameter and the likelihood functions:

$$P\left(\hat{\mathbf{V}}^b|\mathbf{V}, \Sigma^b, \right) \sim N\left(\mathbf{V}, \Sigma^b\right),$$

$$\text{and} \quad P\left(\hat{\mathbf{V}}^a|\mathbf{V}, \theta^c, \theta^m, \Sigma^a\right) \sim N\left(\mathbf{c} + \mathbf{m}\mathbf{V}, \Sigma^a\right) .$$

Stage 2: In the second stage, the statistical model for the underlying true process is used. Thus based on the conditional distribution given the parameters θ^Y:

$$P\left(\mathbf{V}|\theta^Y, \theta^Z\right) \sim N\left(\mathbf{Y}, \sum_{j=1}^{p} \mathbf{T}_j \otimes \mathbf{R}_j\right) ,$$

and priors for (θ^Y, θ^Z), the posterior distribution is obtained for the parameters in the physical deterministic model and for the parameters in the stochastic model.

Stage 3: Conditioned on values of $\mathbf{\Theta}$ updated in the previous two stages, values of $\mathbf{V}$ are simulated using the model for the unobserved true process, at the n_a locations and times of the data $\hat{\mathbf{V}}^a$ and the n_b locations and times of the data $\hat{\mathbf{V}}^b$.

Prior Specification:

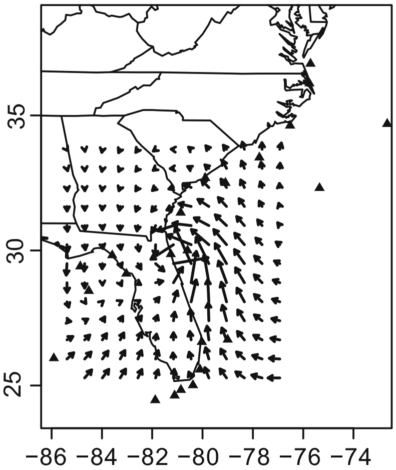
Prior distributions for each parameter complete the Gibbs Sampling algorithm. Prior specification will depend on the form of the bias functions, physical parametrization of the deterministic model and the form of the spatial–temporal covariance functions. Hyperparameters are chosen based on past studies or scientific knowledge. For example, reported information on instrumental measurement error from monitoring networks may be incorporated at this stage. Also the parameters of the deterministic model often have physical constraints that will dictate the form of the prior distributions for θ^Y.

Key Applications

This Bayesian framework for statistical modeling may be applied to a variety of spatial and temporal data sets. At each stage, decisions about model specification (physical and statistical parametrization), prior distributions, appropriateness of assumptions, etc. must be made based on the particular application. The example below describes the model fitting procedure step by step for combining hurricane surface wind information from buoys, satellites, and output from a physics-based model. These estimated wind fields are used as inputs for a numerical coastal ocean model to generate storm surge forecasts for the Eastern U.S. coast. The analysis of the wind data provides one typical application but the methodology for multivariate spatial temporal processes is general enough to be applied to a variety of applications for air quality modeling, weather forecast models, ecological models, etc.

Fusing Buoy and Satellite Wind Data for Hurricane Charley Case Study

For applications in hurricane prediction, surface wind fields are the primary forcing for numerical coastal ocean models. These numerical models simulate the coastal ocean response to the high winds and low pressure associated with hurricanes, such as the height of the storm surge and the degree of coastal flooding. These forecasts can be used for assessments of warnings and evacuation notices. They also provide valuable information for recovery operations and response planning to assess the extent of storm surge inundation, and allow recovery efforts to be organized immediately after a hurricane has made landfall according to the areas most impacted by the storm [10]. Observed data provided by the National Oceanic and Atmospheric Administration (NOAA) are used to predict the hurricane surface wind fields at high temporal and spatial resolution for Hurricane Charely, 2004. The wind measurements are multivariate variables since they are vector values described by a wind speed and direction, that are then decomposed into two orthogonal components denoted as (u, v). The case study of Hurricane Charley is of interest because the hurricane surface winds induced significant



Hurricane Wind Fields, Multivariate Modeling, Figure 2 Buoy locations (shown as triangles) and a sample of HRD wind data for Hurricane Charley, August 14th, 0430 UTC

storm surge along the coast of Georgia and the Carolinas. The best available real-time analyses of hurricane surface wind observations are provided by NOAA's Hurricane Research Division (HRD). These wind fields are a combination of surface weather observations from ships, buoys, coastal platforms, surface aviation reports, reconnaissance aircraft data, and geostationary satellites which have been processed to conform to a common framework for height, exposure and averaging period. Gridded wind fields are used for 6 times on August 14th, 2004, leading up to Hurricane Charley's landfall in South Carolina: 0200, 0430, 0730, 1030, 1330, and 1630 Coordinated Universal Time (UTC). For computational reasons, the gridded wind fields are subsampled to reduce the number of grid points to 225. Wind speed and direction observations are also available from over 20 buoy locations along the Eastern coast from NOAA's National Data Buoy Center (NDBC). For example, Fig. 2 shows the buoy locations and a sample of HRD wind data for a time when Hurricane Charley has just crossed the Florida peninsula. Further details on the data used in this application can be found in [3].

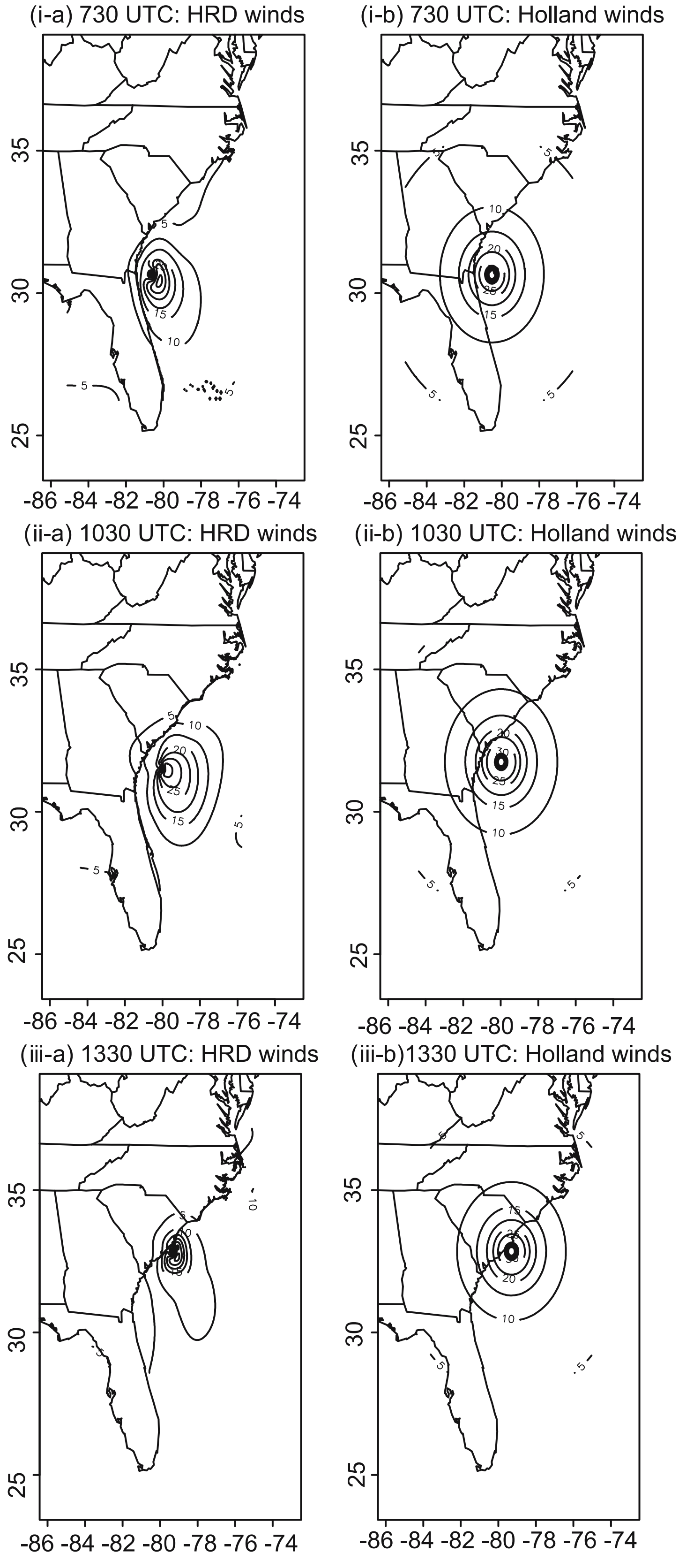
Estimation of Multivariate Wind Model

While the HRD data are spatially dense they are only available approximately every three hours. The buoy data provide wind observations when no new HRD data are available. In addition, the buoy data set is used to estimate potential bias in the HRD winds. Let $\mathbf{V}^a(\mathbf{s}, t) = (v^a(\mathbf{s}, t), u^a(\mathbf{s}, t))^{\mathrm{T}}$ be the u and v components provided by the HRD analysis. Similarly, let $\mathbf{V}^b(\mathbf{s}, t) = (v^b(\mathbf{s}, t), u^b(\mathbf{s}, t))^{\mathrm{T}}$ be wind data from the buoys (after the wind speed and direction are converted to the orthogonal u and v components). Past studies suggest that the magnitude of the bias in the HRD data, which includes satellite and ship data, is larger than the potential bias in the buoys. Validation studies have shown that the HRD wind analyses have estimated errors of up to 10–20% [10]. For this reason although buoy observations are used to create the HRD wind fields, observed winds from the NDBC buoy network are used as a second data source to provide an estimate of the bias in the HRD analysis fields. This approach accounts for the difference in the uncertainty associated with the two data sets.

Exploratory analysis is used to validate that the normality assumption of the statistical model is a reasonable assumption for this setting. Since the buoy data is temporally rich but spatially sparse, the bias function is treated as constant over space, but changes over time: $\mathbf{c}(\mathbf{s}_k, t_l) = (c_u(t_l), c_v(t_l))^{\mathrm{T}}$, for $k = 1, \ldots, r$. Based on exploratory analysis, comparing the buoy and HRD data, multiplicative bias is neglected (e. g. $\mathbf{m} = \mathbf{I}$). Uniform priors are used for the measurement error random effects such that $\sigma_{a_u}, \sigma_{a_v} \sim \text{Uniform}(.01, 5)$ and $\sigma_{b_u}, \sigma_{b_v} \sim \text{Uniform}(.01, 1)$. The relatively informative priors for the buoy data are based on information reported by the NDBC on the accuracy of the buoy equipment. The bias parameters are assigned normal priors with $c_u(t_l)$ and $c_v(t_l)$ distributed $N(0, 1000)$, (1000 is the variance), for each time t_l, $l = 1, \ldots, 6$.

For operational storm surge prediction, the deterministic Holland wind formula is used to force the numerical storm surge model developed for the Eastern US coast [13]. The Holland model is a function of the hurricane central pressure and location of the eye of the storm. The unknown forcing parameters in the model, B and $Rmax$ are typically treated as fixed values based on previous studies or diagnostic runs of the ocean model. Here, the observational data are used to to calculate parameter estimates and standard errors for B and $Rmax$. These estimates are allowed to change over time as the hurricane evolves and more data become available. The Holland formula incorporates important physical information about an individual hurricane provided in real time by the National Hurricane Center (NHC), e. g. [20]. However this is still an idealized model for the surface wind structure. In particular, hurricane winds are often stronger on the right hand side of the hurricane (with respect to the storm movement) rather than symmetric around the storm center as the Holland formula specifies. Also, the organization of the hurricane as it moves along the coastline is very different from the structure in the open ocean, which is not accounted for in the Holland model.

Figure 3 shows contour plots of the wind speed (m/s) for Hurricane Charley at 3 times on August 14th, 2004. The



Hurricane Wind Fields, Multivariate Modeling, Figure 3 Contours of surface wind speed (m/s) for Hurricane Charley, August 14th, 2004 at (i) 730 UTC (ii) 1030 UTC (iii) 1330 UTC. Plots on the left are based on observed winds, plots on the right are based on the Holland model output with $B = 1.9$ and $Rmax = 46$

H

plots on the left are based on observed wind from the HRD at 0730 UTC, 1030 UTC and 1330 UTC. The plots on the right show the Holland model output using $B = 1.9$ and $Rmax = 46$ km. The HRD reports that more than eighty-five percent of storm surge is caused by winds pushing the ocean surface ahead of the storm. Since the symmetric Holland model is overestimating the wind speeds along the coastline this will result in overestimates of the storm surge in these areas. Thus the stochastic component, **Z**, is used to model the variability of the observed data that is not well represented by the deterministic Holland formula. For estimation at Stage 2, priors for the Holland function parameters, $B \sim$ Uniform(1,2.5), $Rmax \sim$ Uniform(15,55), are specified based on physical constraints [9]. Priors are also specified for θ^Z. In this application, ρ_j are stationary correlation functions with parameters ϕ_j; $j = 1, 2$. The HRD wind observations were found to be temporally independent once the Holland mean is subtracted. Recall that the HRD winds are at three hour intervals. Between these times the strength and structure of the hurricane can change drastically. These three–hour periods are treated as windows of spatial–temporal stationarity. A statistical model is fit using all available buoy and HRD data within a three hour window. Furthermore the space–time structure is allowed to change between different windows. An exponential spatial correlation function is estimated such that $\rho_j(h; \phi_j(t_l)) = exp(-h/\phi_j(t_l)), j = 1, 2$ for $h = ||\mathbf{s}_k - \mathbf{s}_{k'}||$ with the range parameters $\phi_j(t_l)$, $l = 1, \ldots, 6$. These ranges are assigned non informative uniform priors, Uniform(10,1000), based on the size of the domain (in km) for the data.

Finally, priors must be specified for the weight matrix, **A**. A conditional modeling approach is used in order to ease the computational cost of working with the full $Np \times Np$ cross–covariance matrix in Stage 2 of the algorithm [15]. Following a standard multivariate normal result:

$$\mathbf{V}_1 \sim N(\mathbf{Y}_1, a_{11}^2 \mathbf{R}_1) \quad (16)$$

$$\mathbf{V}_2|\mathbf{V}_1 \sim N\left(\mathbf{Y}_2 + \frac{a_{21}}{a_{11}}(\mathbf{V}_1 - \mathbf{Y}_1), a_{22}^2 \mathbf{R}_2\right) \quad (17)$$

where $\mathbf{V}_1$ is a vector of the true, unobserved u wind component and $\mathbf{Y}_1$ is the Holland model output for u. Similarly, $\mathbf{V}_2$ and $\mathbf{Y}_2$ correspond to values for the v component. Under the new conditional parametrization the scale parameters a_{11}^2 and a_{22}^2 are assigned inverse gamma distributions ($\frac{1}{a_{11}^2}, \frac{1}{a_{22}^2} \sim Gamma(.01, .01)$) and the weight parameter $\frac{a_{21}}{a_{11}} \sim N(0, 1000)$.

The Gibbs algorithm is implemented within the software GeoBugs [16] to estimate a multivariate statistical model that includes the Holland function and a stochastic component. **Z** is modeled with a separable LMC cross-covariance such that the parameters of the underlying processes are assumed to be equal (i. e. $\phi_1 = \phi_2$). Table 1 shows the posterior distribution summaries of the parameters of the Holland model. The parameter estimates are significantly different for different time periods. Standard values for the Holland model are $B = 1.9$ and $Rmax = 35$ to 55 km depending on the size and intensity of the storm [11]. The estimated values for *Rmax* in this case study do not even fall within this range. Using the standard values greatly overestimates the strength and extent of the winds, especially at times when the storm is moving over the Florida peninsula or just off-shore and is thus less organized. Table 1 also gives the posterior summaries of the bias parameters, c_u and c_v. In general the bias in the HRD winds relative to the buoy observations tends to be less than 1 m/s. The largest bias is seen in the North–South winds at hours 1030 and 1330 UTC.

Hurricane Wind Fields, Multivariate Modeling, Table 1 Posterior medians and 2.5 and 97.5 percentiles for the Holland function and bias parameters based on the separable LMC spatial model for u and v [3]

	B			Rmax		
UTC	2.5%	Median	97.5%	2.5%	Median	97.5%
200	1.62	2.29	2.50	5.00	5.10	5.90
430	1.04	1.35	1.83	5.86	7.11	8.69
730	1.27	1.65	2.22	20.65	23.39	28.43
1030	1.00	1.35	2.45	6.65	12.96	27.44
1330	1.00	1.13	1.60	7.46	10.84	27.45
1630	1.46	1.91	2.45	5.41	7.25	10.87

	c_u			c_v		
UTC	2.5%	Median	97.5%	2.5%	Median	97.5%
200	−.56	−.12	.33	−.83	−.31	.19
430	.38	.76	1.14	−.86	−.02	.35
730	−.07	.44	.89	−1.58	−.95	−.28
1030	−.28	.27	.73	1.45	1.85	2.32
1330	.92	1.41	1.86	1.19	1.62	1.99
1630	−.33	.29	.89	−1.3	−.91	−.45

Storm Surge Prediction

The Princeton Ocean Model (POM) [2] is used for coastal ocean modeling. POM is a fully three–dimensional numerical model and has the capability of modeling the evolution of ocean processes at many layers. In addition, POM uses a bottom following sigma coordinate system to account for differences in water depth. The numerical ocean model used in this study has been developed specifically for modeling along the Eastern United States coast. Unlike global and other large scale forecasting models, this model can be applied at a higher resolution to a much smaller regional domain. As a result, this version of POM is able to predict storm surge with greater accuracy than traditional depth–averaged storm surge models, e. g. see [13,14]. For the case study presented here POM is used to simulate the coastal ocean response to strong hurricane forcing conditions along the coast of the Carolinas and Georgia.

To run the numerical ocean model for the storm surge application, predicted wind fields are required for every grid point of the ocean model domain for 96 ten-minute time increments over a period from 200 UTC to 1750 UTC on August 14, 2004. For this application the ocean model is run at a resolution of 2 minute longitude by 2 minute latitude grid ($\approx$ 3.1km by 3.7km), with a total of 151×115 grid points. A real–time forecasting scenario requires that the prediction methods for these wind fields are computationally efficient. Thus, there must be a balance between the sophistication of any proposed methodology and computing demands of the method. It is computationally infeasible to use the fully Bayesian solution for the posterior predictive distribution, $P(\mathbf{V}(\mathbf{s}_o, t)|\hat{\mathbf{V}})$ to predict the winds at a such a high resolution in real time. Instead an empirical Bayesian approach is implemented and the posterior median values of the parameters, $\hat{\boldsymbol{\Theta}}$, are used as estimates for the fixed "true" parameter values. Then the predictions are made following the conditional distribution, $P(\mathbf{V}(\mathbf{s}, t)|\hat{\mathbf{V}}, \hat{\boldsymbol{\Theta}})$.

Figure 4 shows the observed and predicted v velocities during a time period when Hurricane Charley is entering the ocean model domain for three buoys sites in the region. Similar results were found for the u component. Here we compare two versions of the statistical model. Model 1 uses only the Holland function ($\mathbf{Z}$ is neglected), and estimates the forcing parameters ($B, Rmax$) using the Bayesian statistical framework, rather than using fixed values based on past studies and expert knowledge [13]. Model 2 includes the spatial–temporal stochastic component $\mathbf{Z}$, modeled with a separable LMC model so that the parameters of the underlying processes are assumed to be equal (i. e. $\phi_1 = \phi_2$).

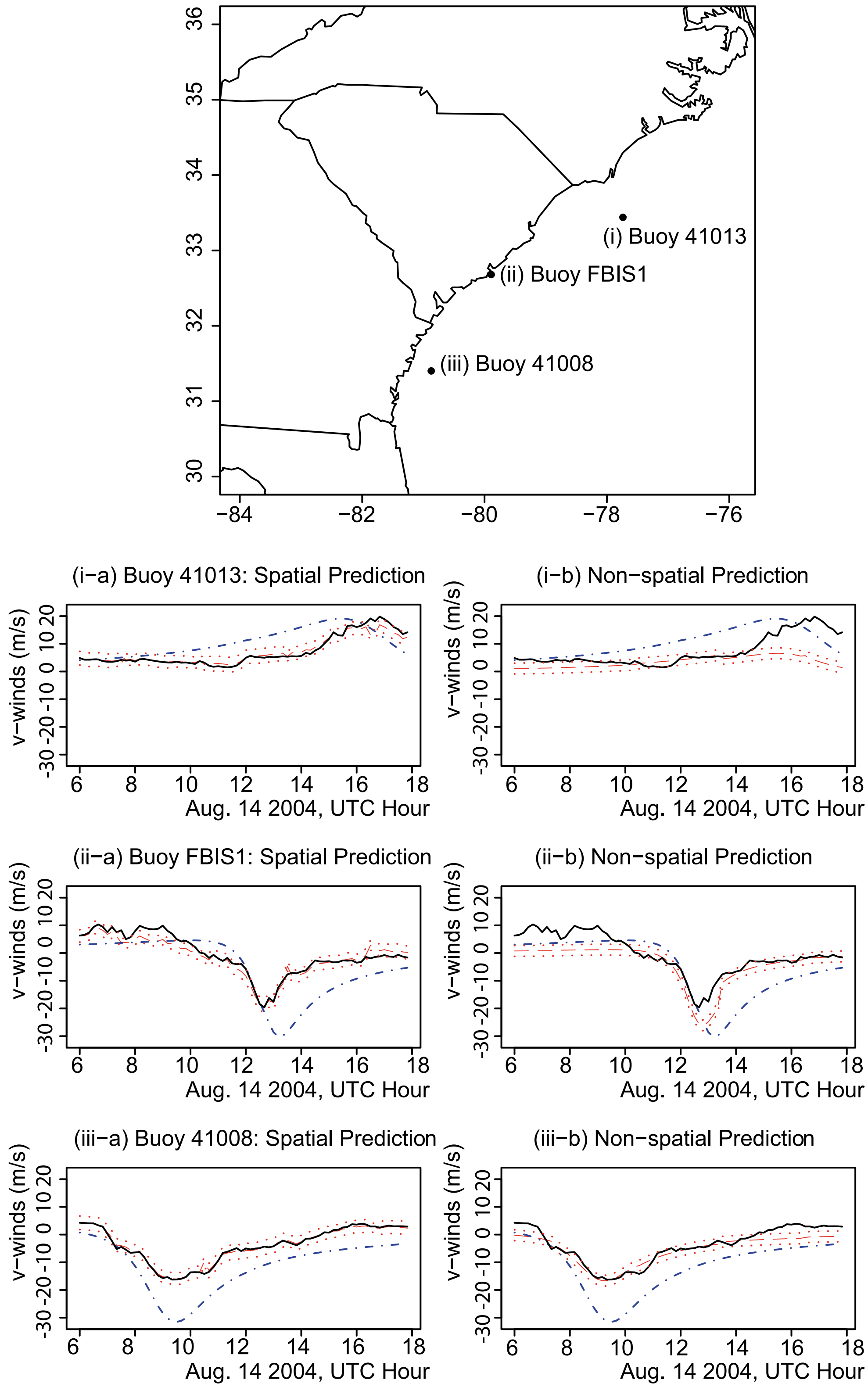
Plots (i-a), (ii-a) and (iii-a) show the predicted values under Model 2 and plots (i-b), (ii-b) and (iii-b) are results based on Model 1. The dashed and dotted line is the output from the Holland model using fixed parameter values. The Holland function is unrealistically smooth and tends to exaggerate the change in the winds as the hurricane approaches these locations. There is also no measure of uncertainty in the values from the Holland function. In contrast, the posterior predicted values (dashed lines) match closely to the observed data at each location. The corresponding 95% posterior predictive intervals are also included as dotted lines but note that since an empirical Bayes approach is used here, these intervals do not account for the uncertainty in the estimation of the model parameters. The non-spatial model, Model 1, also shows an improvement over the Holland model output but the prediction intervals do not capture the observed values as often as the intervals based on the spatial model. Thus there is a clear advantage in using the statistical framework to estimate the Holland model parameters and the uncertainty in these parameters. Furthermore, there is added value in incorporating a multivariate spatial covariance through the stochastic component.

Figure 5 shows a contour plot of mean wind speed values (m/s) based on the predicted u and v winds at all of the ocean model grid points using Model 2. Unlike the Holland output for this time, the predicted winds are able to capture asymmetry in the surface wind field. The plot is for August 14, 1320 UTC. At this time there are only buoy data available (the next available HRD data are at 1330 UTC, shown in Fig. 3). The right hand plot shows the estimated standard errors for the v winds at this time (the standard errors for the u winds are almost identical). The lowest standard errors occur at the locations of the three buoys that are within the ocean model domain. The standard errors tend to range from 1 to 3.5 m/s.

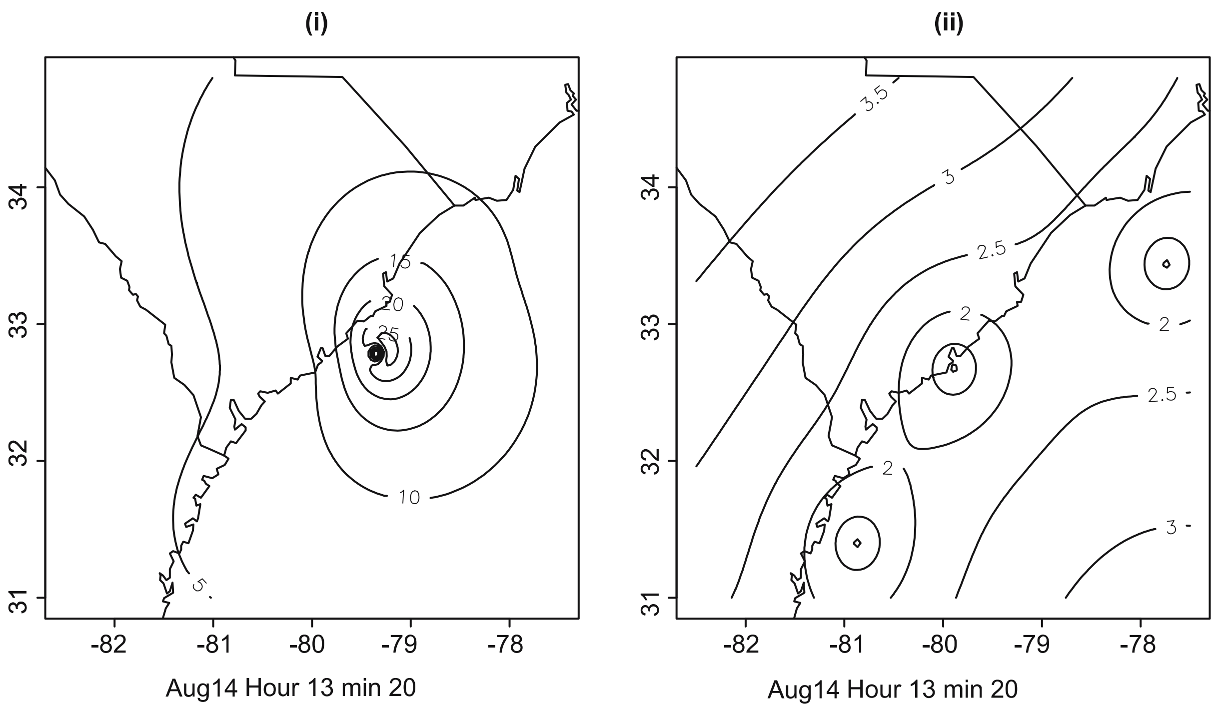
The predicted wind fields are used as the input fields for the POM model to spin up and force the ocean model for a time between August 14th hour 0 UTC and hour 18 UTC. Figure 6 shows the observed change in water levels at each of these sites as Hurricane Charley moves along the coast. The Sunset Beach site shows the greatest storm surge of 1.67 meters. The dashed lines with crosses show the predicted water elevation at each site based on hourly output of the Princeton Ocean Model under the original Holland wind forcings. Using the Holland winds with fixed parameter values the ocean model tends to greatly overestimate the storm surge at three of the four locations. The dotted line shows the predicted values using the statistically based wind estimates, $\mathbf{V} = \mathbf{Y} + \mathbf{Z}$, where $\mathbf{Z}$ is based on Model 2 (separable LMC). The prediction error is reduced and at several of the sites these estimates tend to better capture the timing and magnitude of the peak storm surge. However the model output based on the LMC winds underestimates the greatest surge seen at the Sunset Beach site at 1600 UTC and in fact the Holland model predicts better this peak value.

Diagnostics

Cross validation is used to evaluate alternative model specifications and for comparison of the empirical Bayesian



Hurricane Wind Fields, Multivariate Modeling, Figure 4 Map of 3 NBDC buoys within the POM domain. The remaining plots show the observed *v* winds in m/s at each buoy in solid black for Aug. 14th 200–1750 UTC. The dashed and dotted line is the Holland output based on fixed parameter values ($B = 1.9$, $Rmax = 46$). The dashed lines are the predicted values and the dotted lines are 95% posterior predictive intervals using Model 2 [3]



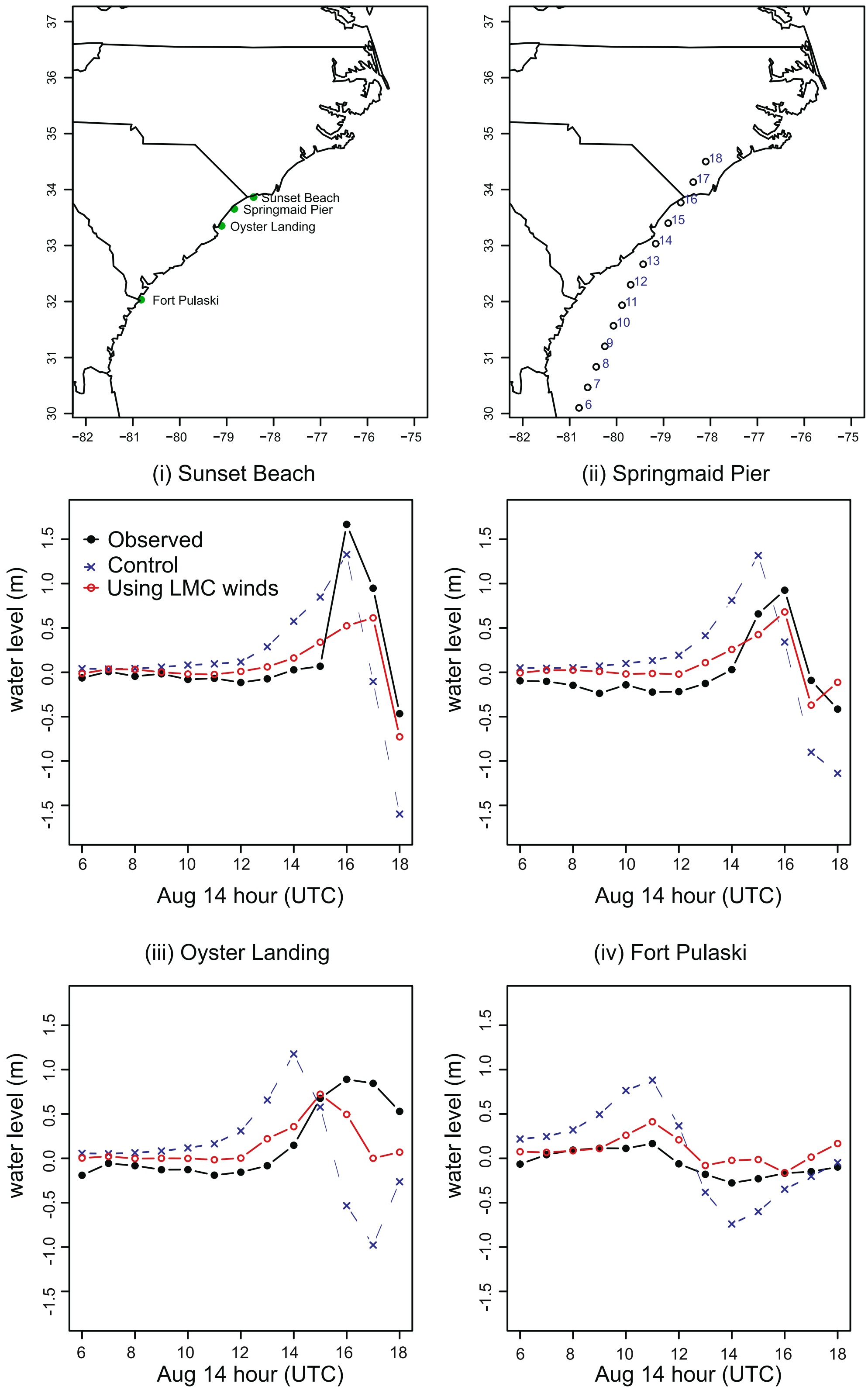
Hurricane Wind Fields, Multivariate Modeling, Figure 5 (i) Contour plot of mean wind speed (m/s) for August 14, 1320 UTC. (ii) Standard error plot for the *v* winds for the same time

estimation to the fully Bayesian solution. Two alternatives to Model 2 are considered. Model 3 uses a non-separable LMC specification for the stochastic component ($\phi_1 \neq \phi_2$) and Model 4 treats the u and v components as independent ($a_{21} = 0$). Models 1, 2, 3 and 4 are used to predict the wind components at thirty HRD grid points that were not used in the model fitting. The median values of the full posterior predictive distribution and the median values of the conditional distribution under the empirical Bayes approach are used to compare to the observed values at the hold out sites. The calibration percents for the different models for the cross validation data are compared. For the statistical model to we well calibrated, the observed values should fall outside the estimated 95% credible prediction intervals less the 5% of the time. All of the spatial models have calibration percents greater than or equal to 95% at all times. However the calibration percents for Model 1 (the non–spatial model) are lower, ranging from 86.7% to 96.7%. Results show that the coverage is almost identical for both the fully Bayes and empirical Bayes prediction. Also, on average, the widths of the fully Bayesian credible intervals are only 4% larger than the intervals based on the empirical approach.

Table 2 shows the root mean square prediction error (RMSPE) values in meters per second for the cross validation for each time period for Models 1 and 2. The Holland model output with fixed parameters ($B = 1.9$, $Rmax = 46$) is compared to the non-spatial model and the separable LMC model using both the fully Bayesian and empirical Bayesian approaches. The Bayesian estimation of the Holland parameters decreases the RMSPE values by more than forty percent in most cases. The predictions based on the spatial model have much lower RMSPE values than the non-spatial model for all time periods and the values are very similar for both estimation methods. Hence, the empirical Bayes prediction is found to be a reasonable approach in this case, as well as being significantly more computationally efficient.

Hurricane Wind Fields, Multivariate Modeling, Table 2 RMSPE for **V** (m/s) based on 30 cross validation sites using the Holland model output (fixed parameters), fully Bayesian prediction and the empirical Bayesian approach based on the non–spatial model (Model 1) and separable LMC model (Model 2)

Model	200	430	730	1030	1330	1630
Holland Output	9.57	7.70	6.33	7.21	6.53	6.40
Full Bayes (Non–spatial)	3.62	3.30	3.51	4.55	3.66	3.05
Full Bayes (Spatial)	.54	.25	.84	1.79	.64	.18
Emp. Bayes (Spatial)	.51	.43	1.13	1.84	.62	.17



Hurricane Wind Fields, Multivariate Modeling, Figure 6 Maps show the location of 7 coastal water elevation gauges and the track of Hurricane Charley on August 14th for hours 6 UTC to 18 UTC. Time series plots show observed water elevation at each site as a solid black dotted line. The dotted line(open circles) shows the predicted elevation using the predicted winds from the LMC statistical model to force the ocean model run. The dashed line with crosses shows the estimates when the ocean model is forced using the original Holland wind model [3]

Future Directions

Here a stochastic component was used to capture spatial–temporal variability in the residual wind fields (i. e. true winds minus Holland output). This approach to wind field modeling provides a cohesive framework for obtaining parameter estimates and standard errors, as well as estimated wind fields through spatial prediction. Another approach would be to introduce a more sophisticated deterministic wind model. A coupled atmospheric-oceanic numerical model can be used to simulate the surface winds at the boundary layer of the ocean model. However the computation time required to produce these modeled winds at high enough resolution for coastal prediction (1 to 4 km grids) prevents such model runs from being used in real-time applications. This motivates the problem of defining a multivariate spatial–temporal model to capture the variability in the wind fields not accounted for in the Holland wind formulation. Assuming that the processes of interest are Gaussian, the flexible multivariate model is specified by a cross-covariance function that is valid by construction. Alternatively, the multivariate spatial data using a Markov random field approach, such as the multivariate conditionally autoregressive (MCAR) model described in [8]. The methods proposed here rely heavily on the assumption that the data are normally distributed. This is an appropriate assumption for the residual wind fields but may not apply in other applications.

Acknowledgements

This study was supported by the Carolina Coastal Ocean Observation and Prediction System (Caro-COOPS) project under NOAA (Grant No. NA16RP2543). Funding was also provided by the National Science Foundation (Grant No. DMS 0353029).

Cross References

► Autocorrelation, Spatial
► Data Analysis, Spatial
► Hierarchical Spatial Models
► Semivariogram Modeling
► Uncertainty, Modeling with Spatial and Temporal

Recommended Reading

1. Banerjee, S., Carlin, B.P., Gelfand, A.E.: Hierarchical modeling and analysis for spatial data. Chapman & Hall, New York (2004)
2. Blumberg, A.F., Mellor, G.L.: A description of a three-dimensional coastal ocean circulation model. In: Heaps, N.S. (ed.) Three-Dimensional Coastal Ocean Models, pp. 1–16. American Geophysical Union, Washington, D.C. (1987)
3. Foley, K.M., Fuentes, M.: A statistical framework to combine multivariate spatial data and physical models for hurricane surface wind prediction. JABES, to appear
4. Fuentes, M.: A high frequency kriging approach for non-stationary environmental processes. Environmetrics **12**, 469–483 (2001)
5. Fuentes, M., Chen, L., Davis, J.M., Lackmann, G.M.: Modeling and predicting complex space-time structures and patterns of coastal wind fields. Environmetrics **16**, 449–464 (2005)
6. Fuentes, M., Raftery, A.: Model evaluation and spatial interpolation by Bayesian combination of observations with outputs from numerical models. Biometrics **61**, 36–45 (2005)
7. Gelfand, A.E., Schmidt, A.M., Banerjee, S., Sirmans, C.F.: Nonstationary multivariate process modeling through spatially varying coregionalization (with discussion). TEST **13**, 1–50 (2004)
8. Gelfand, A.E., Vounatsou, P.: Proper multivariate conditional autoregressive models for spatial data analysis. Biostatistics **4**, 11–25 (2003)
9. Holland, G.: An analytic model of the wind and pressure profiles in hurricanes. Monthly Weather Review **108**, 1212–1218 (1980)
10. Houston, S.H., Shaffer, W.A., Powell, M.D., Chen, J.: Comparisons of HRD and SLOSH surface wind fields in hurricanes: Implications for storm surge modeling. Weather and Forecasting **14**, 671–686 (1999)
11. Hsu, S.A., Yan, Z.: A note on the radius of maximum wind for hurricanes. J. Coastal Res. **14**, 667–668 (1998)
12. Kibria, B.M.G., Sun, L., Zidek, J.V., Le, N.D.: Bayesian spatial prediction of random space-time fields with application to mapping $PM_{2.5}$ exposure. J. Am. Stat. Assoc. **97**, 112–124 (2002)
13. Peng, M., Xie, L., Pietrafesa, J.: A numerical study of storm surge and inundation in the Croatan-Albemarle-Pamlico Estuary System. Estuarine, Coastal and Shelf Science **59**, 121–137 (2004)
14. Pietrafesa, L.J., Xie, L., Morrison, J., Janowitz, G.S., Pelissier, J., Keeter, K., Neuherz, R.A.: Numerical modelling and computer visualization of the storm surge in and around the Croatan-Albemarle-Pamlico Estuary system produced by Hurricane Emily of August (1993). Mausam **48**, 567–578 (1997)
15. Schmidt, A.M., Gelfand, A.E.: A Bayesian coregionalization approach for multivariate pollutant data. J. Geophys. Res. **108**, 8783 (2003)
16. Spiegelhalter, D., Thomas, A., Best, N., Gilks, W.: Bugs.5 Bayesian inference using Gibbs sampling. Manual, version ii. MRC Biostatistics Unit, Institute of Public Health, Cambridge, UK (1996)
17. Wackernagel, H.: Multivariate geostatistics: An introduction with applications. Springer-Verlag, New York (2003)
18. Wikle, C.K.: Hierarchical models in environmental science. Int. Stat. Rev. **71**, 181–199 (2003)
19. Wikle, C.K., Milliff, R.F., Nychka, D., Berliner, L.M.: Spatiotemporal hierarchical Bayesian modeling: Tropical ocean surface winds. J. Am. Stat. Assoc. **96**, 382–397 (2001)
20. Xie, L., Bao, S., Pietrafesa, L., Foley, K., Fuentes, M.: A real-time hurricane surface wind forecasting model: Formulation and verification. Monthly Weather Review **134**, 1355–1370 (2006)

Hydrogeology

► Distributed Hydrologic Modeling
► Hydrologic Impacts, Spatial Simulation

H

Hydrologic Impacts, Spatial Simulation

GRAEME AGGETT[1], CHRIS MCCOLL[2]
[1] Riverside Technology, Inc., Fort Collins, CO, USA
[2] Department of Geography, Central Washington University, Ellensburg, WA, USA

Synonyms

Hydrologic modeling and hydraulic modeling with GIS; Hydrogeology

Definition

Spatial simulation of future hydrologic impacts involves deterministic or probabilistic modeling approaches that attempt to simulate likely changes in hydrology, and subsequent hydrologic response (impacts) of these changes, for a particular study area. The modeling approach might be focused on understanding the spatial impacts of predicted hydrologic changes (e. g. rainfall intensity), and/or changes in parameters impacting rainfall-runoff response and flow routing (e. g. changing land use). The goal is to produce spatial (map) and other data outputs that can assist planners and managers better understand the spatial ramifications of an uncertain future. Where appropriate and possible, estimates of uncertainty should be embedded in the map output. This information might be used to develop more informed and hence effective land use plans, flood mitigation strategies, or management strategies for habitat. The approach utilizes recent technological advances in the geospatial and hydrologic sciences to develop predictive modeling applications that can help environmental planners and managers better understand the spatial implications of hydrological processes changing in response to issues such as climate change and rapid urban development.

Historical Background

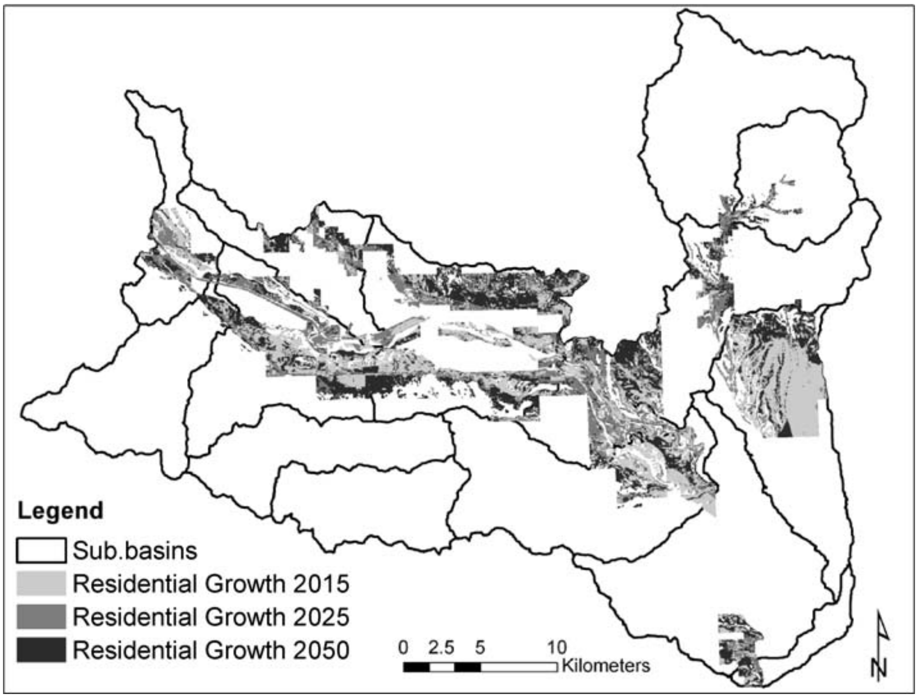
The approach has been stimulated by the development and utilization of spatially distributed parameters for hydrologic modeling brought on by advances in computing and geographic information systems (GIS). GIS have increasingly become a valuable management tool, providing an effective infrastructure for managing, analyzing, and visualizing disparate datasets related to soils, topography, land use, land cover, and climate (Liao and Tim, 1997; Miller, R.C., Guertin and Heilman, 2004). The integration of GIS with hydrologic and hydraulic models as a data pre/postprocessor have simplified data management activities by enabling relatively easy and efficient extraction of multiple modeling parameters at the watershed scale (Ogden et al., 2001).

Some methods for integrating hydrologic models with GIS have been categorized as "loose," "close," or "tight" coupling (Liao and Tim, 1997). Loose coupling methods usually involve data exchange using ASCII or binary data formats. An interface program is normally used to convert and organize GIS data into a format required by the hydrologic or land use model. Advantages of loose integration include ease of development and use with a wide range of commercial GIS software. Close coupling incorporates slight modifications to the control programs in the GIS software, providing improved data transfer between the model and the GIS database. There tends to be overlap between loose and close coupling methods. However, the close coupling method passes information between the GIS and the model via memory-resident data models rather than external files. This enhancement leads to improved model interactions and performance (Di Luzio et al., 2004). Tightly coupled model integration focuses on incorporating the functional components of one system within the other (i. e. the model within the GIS program). The GIS and model are no longer maintained separately. They instead share processes and data in order to reduce the redundancy in development and operation. This approach eliminates the use of interface modules and transfer files, and thereby promotes better system performance (Liao and Tim, 1997).

Initial attempts to link existing hydrologic models with GIS utilized loose coupling, and frequently involved manual as well as automated parameter development. For example, Warwick and Haness (1994) used Arc/Info to determine hydrologic parameters directly for the Hydrologic Engineering Center-1 model (HEC-1), while separate line coverages defining the runoff routing were created manually. Suwanwelarkamtorm (1994) derived semi-distributed hydrologic modeling using GIS for the management of watersheds and assessed the effect of land-use change using an integrated approach with HEC-1 and ILWIS (Integrated Land and Water Information System). The ability of the model to simulate future and past flood hydrographs based on hypothetical future as well as historical land-use conditions were demonstrated. The results of simulation runs demonstrated that for the study area, when forest area was reduced, more runoff would occur in every sub-catchment and also at the outlet.

Scientific Fundamentals

Water resource planners and managers can be required to provide information to facilitate preparation for future flood hazard situations and to develop responsible regula-



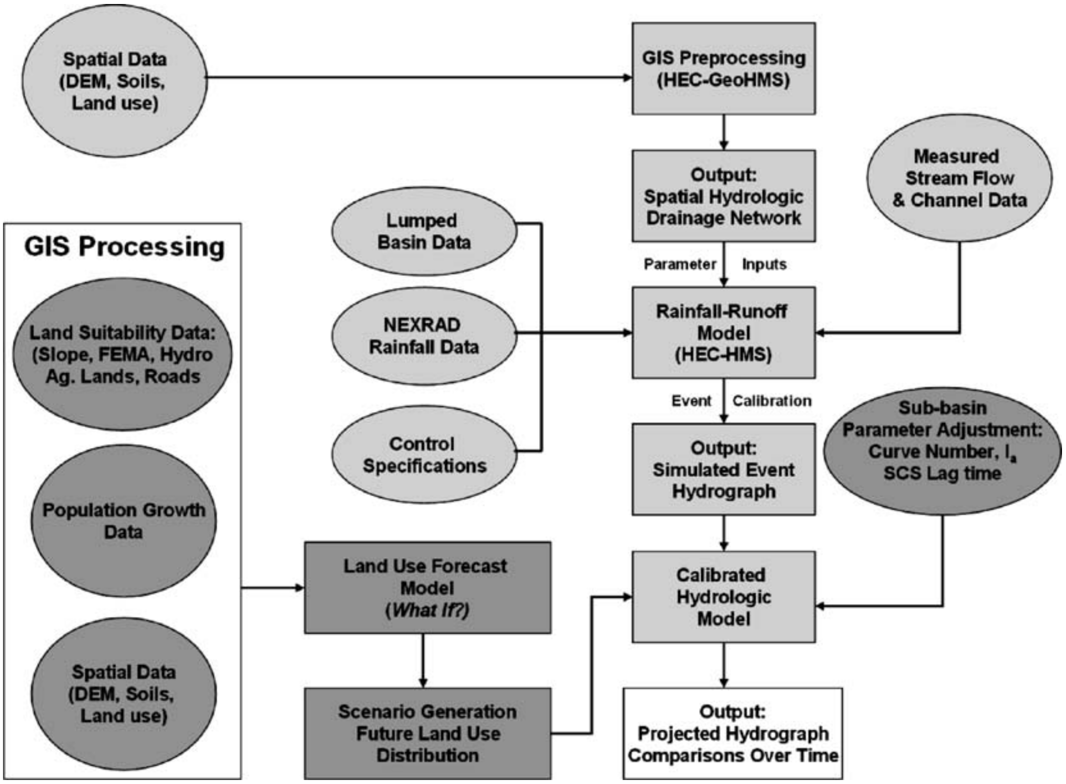
Hydrologic Impacts, Spatial Simulation, Figure 1 Forecasted low density residential growth based on high growth rates for years 2015, 2025, and 2050 (Source: McColl and Aggett, 2007)

tions for sound floodplain development. If already in existence, historical hydrologic flow records, stage, and precipitation records can sometimes satisfy these planning needs. However, in many cases watershed runoff must be predicted to provide the needed information to support these preparatory and regulatory decisions. For example, a flood-damage reduction study may require an estimate of the increased volume of runoff that will result due to proposed changes to land use within a watershed. Unfortunately, no data record is available because the land use change has not yet occurred. Waiting to observe the future hydrologic impacts of proposed land use changes could result in losses of property, and even human life.

An alternative to "waiting and observing" is to use a hydrologic or hydraulic mathematical model to provide the information to aid decision-making processes that may impact future hydrologic responses of a given area (United States Army Corps of Engineers, 2001). Caution is required when applying mathematical models to describe complex hydrologic systems. Models are only approximate representations of these complex natural processes, and usually incorporate the assumptions of model developers who have attempted to define the critical processes occurring within the watershed, and to develop relationships between these processes. Models may involve over-simplifications, or conversely over-specification, of hydrologic processes, which may or may not be valid for other site-specific applications due to the uniqueness of the study area for which they were developed (Kalin and Hantush, 2003). Careful hydrologic model selection is therefore critical.

However, to gain a precise and complete understanding of the hydrologic impacts that land use change may impose upon every location within a watershed would require impossibly high degrees of data collection. This level of data collection would be financially unfeasible at the watershed scale (Santhi et al., 2001). Hydrologic and hydraulic models represent the most practical and realistic means to examine the flow of water on a watershed scale, in a manner allowing planners and managers to determine the effects of different land use scenarios over long time periods (Di Luzio et al., 2004; Santhi et al., 2001). Moreover, the application of hydrologic and hydraulic models allows users to identify high priority areas, valuable information when allocating scarce resources to, for example, flood mitigation efforts (Srinivasan et al., 1998).

A general approach to simulating the spatial variability of hydrologic impacts typically involves a multi-step procedure that might involve several or all of these components: 1) creation of an accurate hydrologic component representation of the watershed area; 2) determination of values for hydrologic modeling parameters and associated calibration for a specific storm, or range of storm events (which may incorporate predicted changes in magnitude and frequency); 3) generation of land use distribution patterns for initial and target years based upon a selected comprehen-



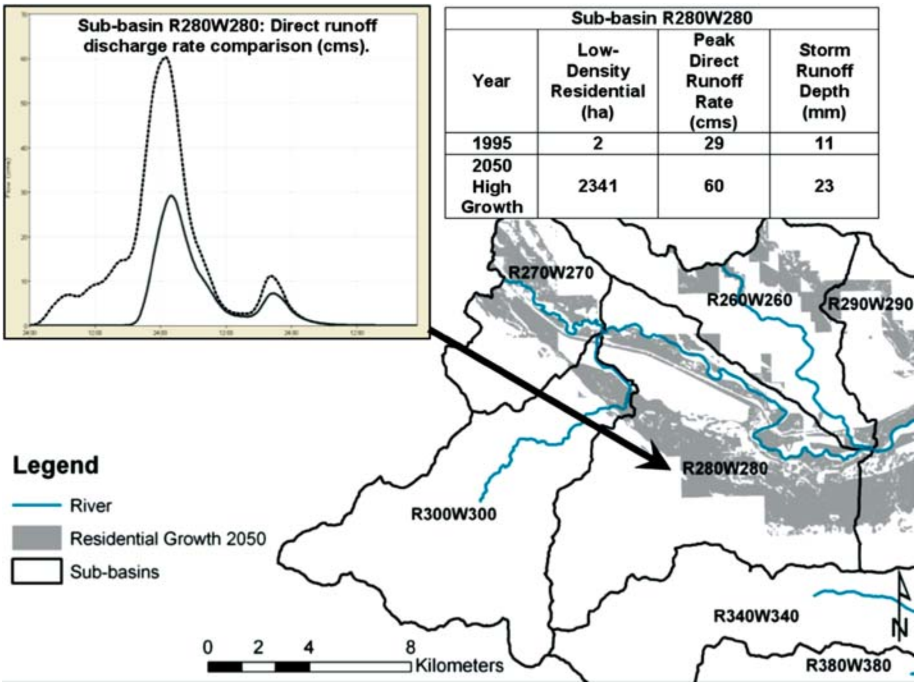
Hydrologic Impacts, Spatial Simulation, Figure 2 Workflow schematic illustrating the integration of hydrologic and land use models (Source: McColl and Aggett, 2007)

sive land use policy, or forestry cover for initial and target years based upon a selected harvesting strategy; 4) use of forecasted land use/forestry cover datasets as inputs into the calibrated hydrologic model to generate storm hydrographs for target time periods; 5) terrain modeling of the receiving channel(s) (using high resolution terrain data, such as LiDAR, where available); 6) mapping of hydrologic/hydraulic data outputs, representing uncertainty when possible.

Key Applications

Worldwide, climate change and climate variability is causing significant and often unpredictable impacts on the hydrological system. In many locations, the risk of and vulnerability to floods is increasing due to changes in rainfall patterns and increased frequency of large events. In others, agricultural industries are threatened by prolonged droughts. Potential impacts may be exacerbated by rapid changes in land cover, development into flood-prone areas as a result of socio-economic pressure, and increasing municipal demands for water traditionally utilized by agriculture.

An increasing amount of research has been focused upon the modeling of future land use change via urbanization and the associated changes in hydrologic/hydraulic responses to precipitation events under increased amounts of urban land uses. In the US, data collected through the National Resources Inventory have shown that from the periods 1982–1992 to 1992–2001, the annual rate of land conversion has nearly doubled (United States Department of Agriculture-Natural Resources Conservation Service, 2003). Studies have revealed that conversions of this type can alter a watershed's response to precipitation events, leading to increased volumes of surface water runoff, greater incidence of flooding, altered downstream river channel geometry through erosion, and the degradation of aquatic habitat of fish and other biota. Recent applications that integrate GIS technologies, land use forecasting models, and hydrologic models (e. g. Beighley et al., 2003; McColl and Aggett, 2006) have presented an opportunity to examine watershed response to precipitation under different future land use patterns reflective of various land use planning policies under consideration. These applications initially develop a hydrologic model that is calibrated to a historical precipitation event. Hydrologic parame-



Sub-basin R280W280			
Year	Low-Density Residential (ha)	Peak Direct Runoff Rate (cms)	Storm Runoff Depth (mm)
1995	2	29	11
2050 High Growth	2341	60	23

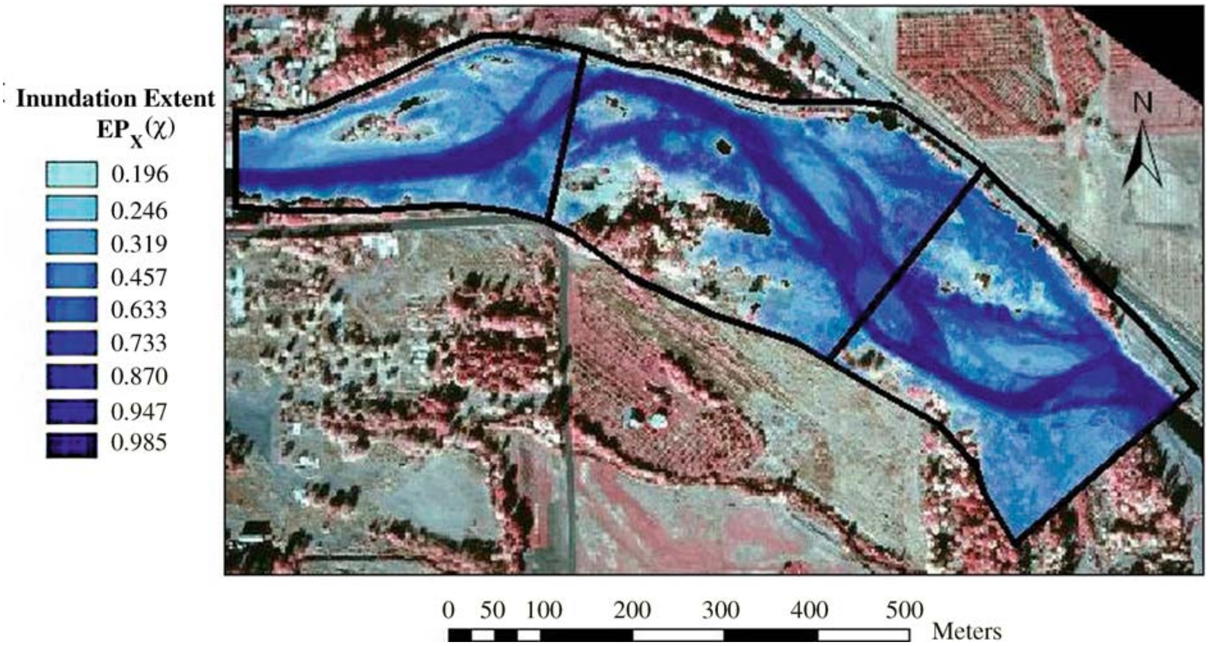
Hydrologic Impacts, Spatial Simulation, Figure 3 Assessment of hydrologic response to forecasted low-density residential growth for a sub-basin, assuming a high growth rate (Source: McColl and Aggett, 2007)

ter data required to run the hydrologic model is obtained for the study area, coinciding with the precipitation event date. These parameters include slope, soil type, land cover type, and river-channel slope and cross-sectional geometry, each of which can be readily extracted from available geospatial datasets (e. g. digital elevation models, digital soil and land cover datasets). River discharge volumes are computed by a hydrologic model at specified points within the stream network, which can then be compared to observed storm event discharge volumes. The ability to compare observed storm event discharge volumes to simulated volumes enables modelers to determine whether the model is reasonably characterizing the storm event and its associated runoff volumes.

Once the hydrologic model is computing accurate river discharge volumes, a land use development or forestry model can be applied to forecast future land use patterns. For example, McColl and Aggett (2006) forecasted development patterns based upon selected comprehensive land use policies and estimated population growth rates derived from census data, county parcel records, and re-zoning trends. Suitable lands are selected through the application of suitability factors assumed to affect growth, such as proximity to transportation arteries, flood plain delineation, riparian corridors, slope hazards, and existing land use types that may be converted to other land uses. The estimated population growth is then allocated to lands that have been classified as suitable for residential growth (Fig. 1).

The forecasted land use patterns can then be integrated with the land use data that was initially used to calibrate hydrologic model. All hydrologic parameters, precipitation inputs, and time frames are held constant, except the parameters whose values are derived from land use type. The hydrologic model is reapplied using the forecasted land use patterns to compute new runoff volumes. Figure 2 illustrates a typical workflow used in this approach. The calculated estimates enable land use managers to determine whether specific land use planning policies significantly alter runoff volumes. These quantifiable results provide land use decision makers with an additional means for evaluating drafted land use plans. Various comprehensive planning policies can be developed and evaluated using this approach to determine which one would incur the least hydrologic impact on the watershed.

Although some applications evaluate discharge volumes strictly at the outlet point of the watershed, the capability for measuring hydrologic impacts of land use change at a subbasin scale can provide more detailed information for decision makers as they evaluate proposed land use policies. For example, hydrologic impacts of land use change over an entire watershed may be insignificant because the watershed as a whole can act as a buffer to localized runoff. However, when measured at specific subbasin discharge points, the runoff volumes and subsequent impact on a localized area can be quite substantial (e. g. Fig. 3). In light of this information, land use managers may be prompted to alter proposed or existing



Hydrologic Impacts, Spatial Simulation, Figure 4 Exceedence probability map of flooding on a reach of the Naches River, WA. Source: Swan and Aggett (2007)

land use planning policies in an effort to mitigate for the potential increase in flood risks and the associated negative impacts these computed peak flows could have upon local aquatic habitat. Furthermore, this additional information may provide decision makers with the impetus to examine other questions more comprehensively regarding future flood risks, such as: "Do existing downstream flood mitigation structures have the appropriate structural design to safely accommodate the forecasted increases in precipitation runoff being generated for each specific subbasin?"
The model integration concepts described above can be extended to incorporate a river hydraulic model. The hydraulic model can analyze river channel physical characteristics and predict flood inundation scenarios, based upon river discharge hydrographs provided by the hydrologic model. Recent studies by Whiteaker et al. (2006) have demonstrated the ability to integrate models such as HEC-HMS (Hydrologic Modeling System) with the HEC-RAS (River Analysis System) to provide emergency planners and land use planners with predicted flood inundation maps. Hydrographs computed by HEC-HMS were used as river discharge inputs for HEC-RAS. Based upon the discharge hydrographs HEC-RAS was able to generate flood map inundation estimates within a spatial and temporal context. This level of model integration provides land use and emergency planners with additional layers of information enabling them to begin to assess whether existing Flood Insurance Rate Maps are still relevant, considering projected increases in peak discharge.

Future Directions

The main goal of spatial simulation of future hydrologic impacts is to produce maps that can assist planners and managers better understand the potential spatial ramifications of an uncertain hydrologic future. Moglen et al. (2003) have stressed that 'truly intelligent smart growth' ought to be quantifiably superior to other proposed land development plans. However, because there are many data uncertainties embedded in this general approach (including uncertainties in the input hydrometeorologic, hydrologic and terrain data), there is a requirement to develop methods that can convey this uncertainty to limit misuse of map products. This concern is chiefly focused on three demands: (i) a fundamental scientific obligation to describe how close modeled information may be to a 'truth' it is intended to represent; (ii) a requirement that individual and agency reputations be protected, particularly when geographic information is used to support administrative decisions subject to appeal; and (iii) protection against litigation by those who might, in the future, allege to have experienced damage through planning based on map products of deficient quality. Where appropriate and possible, estimates of uncertainty should thus be embedded in the map output. For example, in mapping outputs from various hydraulic models both Legleiter and Goodchild (2005) and Aggett and Wilson (2007) point out the utility of fuzzy logic for incorporating uncertainty. Fuzzy logic addresses one of the fundamental limitations of spatial databases, namely its reliance on either/or logic (Burrough, 1996; Hunter and Goodchild, 1996). Fuzzy logic allows for the existence of transition zones between features using a function that assigns boundary components a grade of membership which corresponds to the degree of similarity between representative components of each feature. de Bruin (2000) used probabilities to determine the membership functions for fuzzy membership classes. Swan and Aggett (2007) utilize this strategy to represent the inundation extent of the modeled flows with some indication of variability. In their research, each individual cell in the depth layer required a membership value based on its depth. In deriving the membership function for the layer, the highest membership values were assigned to the cells containing the deepest inundation depths, centering the function around the maximum depth. In a similar fashion, fuzzy logic can also represent transition zones through some indication of uncertainty or error in the dataset. For example, the membership function could only be applied to areas of questionable credibility. In modeling future flood hazards, these areas might typically be transition zones between features such as the area dividing the wetted and dry extents of the channel.

Cross References

► Data Analysis, Spatial
► Distributed Hydrologic Modeling
► Environmental Planning and Simulation Tools
► Multicriteria Spatial Decision Support Systems
► Rural Hydrologic Decision Support
► Uncertain Environmental Variables in GIS
► Visualizing Constraint Data

Recommended Reading

Aggett, G.R., Wilson, J.P.: Creating and coupling a high-resolution DEM with a 1-D hydraulic model in a GIS for scenario-based assessment of avulsion in a gravel-bed river. Geomorphology, in review (2007)

Beighley, R.E., Melack, M., Dunne, T.: Impacts of California's climatic regimes and coastal land use change on streamflow characteristics. J. Am. Water Resour. Assoc. **39**(6), 1419–1433 (2003)

Burrough, P.A.: Fuzzy mathematical methods for soil survey and land evaluation. J. Soil Sci. **40**, 477–492 (1989)

Burrough, P.A.: Natural Objects with Indeterminate Boundaries. In: Burrough, P.A., Frank, A.U. (eds.) Geographic Objects with Indeterminate Boundaries, pp. 345. Taylor & Francis, Ltd., London (1996)

Burrough, P.A., van Gaans, P.F.M., Hootsmans, R.: Continuous classification in soil survey: spatial correlation, confusion and boundaries. Geoderma, **77**, 115–135 (1997)

de Bruin, S.: Querying probablistic land cover data using fuzzy set theory. Int. J. Geogr. Inf. Sci. **14**(4), 359–372 (2000)

Di Luzio, M., Srinivasan, R., Arnold, J.G.: A GIS-coupled hydrological model system for the watershed assessment of agricultural nonpoint and point sources of pollution. Transactions in GIS **8**(1), 113–136 (2004)

Hunter, G.J., Goodchild, M.F.: Communicating uncertainty in spatial databases. Transactions in GIS **1**(1), 13–24 (1996)

Kalin, L., Hantush, M.M.: Evaluation of sediment transport models and comparative application of two watershed models (U.S. Environmental Protection Agency, National Risk Management Research Laboratory Paper EPA/600/R-03/139). U.S. Environmental Protection Agency, Cincinnati, OH (2003)

Legleiter, C.J., Goodchild, M.F.: Alternative representations of in-stream habitat: classification using remote sensing, hydraulic modeling, and fuzzy logic. Int. J. Geogr. Inf. Sci. **19**(1), 29–50 (2005)

Liao, H., Tim, U.S.: An interactive modeling environment for non-point source pollution control. J. Am. Water Resour. Assoc. **3**(3), 591–603 (1997)

McColl, C., Aggett, G.: Land use forecasting and hydrologic model integration for improved land use decision support. J. Env. Manag. **84**, 494–512 (2007)

Miller, R.C., Guertin D.P, Heilman, P.: Information technology in watershed management decision making. J. Am. Water Resour. Assoc. **40**(2), 349–357 (2004)

Moglen, G.E., Gabriel, S.A., Faria, J.A.: A Framework for Quantitative Smart Growth in Land Development. J. Am. Water Resour. Assoc. **39**(4), 947–959 (2003)

Ogden, F.L., Garbrecht, J., DeBarry, P.A. Johnson, L.E.: GIS and distributed watershed models II: Modules, interfaces, and models. J. Hydrol. Eng. **6**(6), 515–523 (2001)

Santhi, C., Arnold, J.G., Williams, J.R., Hauck, L.M., Dugas, W.A.: Application of a watershed model to evaluate management effects on point and nonpoint source pollution. Transactions of the American Society of Agricultural Engineers **44**(6), 1559–1570 (2001)

Srinivasan, R., Ramanarayanan, R., Arnold, J., Bednarz, S.: Large area hydrologic modeling and assessment part II: Model application. J. Am. Water Resour. Assoc. **34**(1), 91–101 (1998)

Suwanwerakamtorn, R.: GIS and hydrologic modeling for the management of small watersheds. Intern. J. Appl. Earth Obs. Geoinf. (ITC) 1994–4, 343–348 (1994)

Swan, D., Aggett, G.: Two-dimensional modeling of stream channel dynamics in the Naches River, Washington. Transactions in GIS, in review (2007)

United States Army Corps of Engineers: Hydrologic modeling system HEC-HMS: User's manual. USACE Hydrologic Engineering Center, Washington, DC (2001)

Warwick, J.J., Haness, S.J.: Efficacy of ARC/INFO GIS Application to Hydrologic Modeling, J. Water Resour. Plann Manag. **120**(3), 366–381 (1994)

Whiteaker, T.L., Robayo, O., Maidment, D.R., Obenour, D. From NEXRAD rainfall map to an inundation map. J. Hydrol. Eng. **11**(1), 37–45 (2006)

Hydrologic Modeling and Hydraulic Modeling with GIS

► Hydrologic Impacts, Spatial Simulation

Hydrology

► Distributed Hydrologic Modeling

Hypothesis Validation in Spatial Data

► Geographic Knowledge Discovery

Identity Aware LBS

► Privacy Threats in Location-Based Services

Identity Unaware LBS

► Privacy Threats in Location-Based Services

iDistance Techniques

H.V. JAGADISH[1], BENG CHIN OOI[2], RUI ZHANG[3]
[1] Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA
[2] Department of Computer Science, National University of Singapore, Singapore, Singapore
[3] Department of Computer Science and Software Engineering, The University of Melbourne, Parkville, VIC, Australia

Synonyms

Query, nearest neighbor; Scan, sequential

Definition

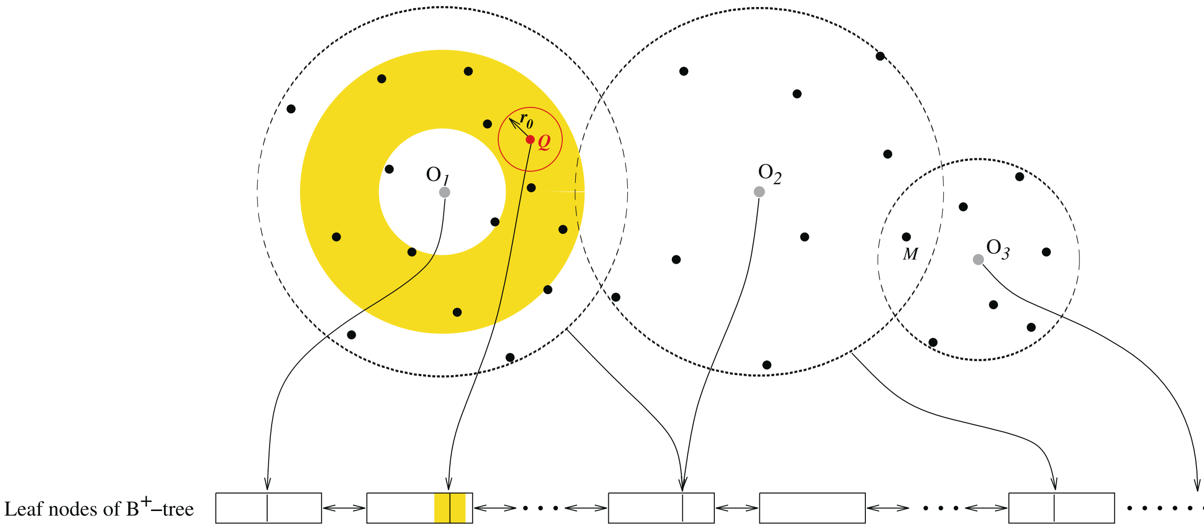
The iDistance is an indexing and query processing technique for k nearest neighbor (kNN) queries on point data in multi-dimensional metric spaces. The kNN query is one of the hardest problems on multi-dimensional data. It has been shown analytically and experimentally that any algorithm using hierarchical index structure based on either space- or data-partitioning is less efficient than the naive method of sequentially checking every data record (called the *sequential scan*) in high-dimensional spaces [4]. Some data distributions including the uniform distribution are particularly hard cases [1]. The iDistance is designed to process kNN queries in high-dimensional spaces efficiently and it is especially good for skewed data distributions, which usually occur in real-life data sets. For uniform data, the iDistance beats the sequential scan up to 30 dimensions as reported in [3]. Building the iDistance index has two steps. First, a number of reference points in the data space are chosen. There are various ways of choosing reference points. Using cluster centers as reference points is the most efficient way. Second, the distance between a data point and its closest reference point is calculated. This distance plus a scaling value is called the point's *iDistance*. By this means, points in a multi-dimensional space are mapped to one-dimensional values, and then a B^+-tree can be adopted to index the points using the iDistance as the key. A kNN search is mapped to a number of one-dimensional range searches, which can be processed efficiently on a B^+-tree. The iDistance technique can be viewed as a way of accelerating the sequential scan. Instead of scanning records from the beginning to the end of the data file, the iDistance starts the scan from spots where the nearest neighbors can be obtained early with a very high probability.

Historical Background

The iDistance was first proposed by Cui Yu, Beng Chin Ooi, Kian-Lee Tan and H. V. Jagadish in 2001 [5]. Later, together with Rui Zhang, they improved the technique and performed a more comprehensive study on it in 2005 [3].

Scientific Fundamentals

Figure 1 shows an example of how the iDistance works. The black dots are data points and the gray dots are reference points. The number of reference points is a tunable parameter, denoted by N_r. The recommended value for N_r is between 60 and 80. In this example, $N_r = 3$. At first, 3 cluster centers of the data points, O_1, O_2, O_3 are identified using a clustering algorithm, and these cluster centers are chosen as reference points. Each data point p is assigned to its closest reference point and hence all the data points are divided into 3 partitions P_1, P_2 and P_3. Let d_i be the distance between O_i and the farthest data point to O_i in P_i. Then the range of P_i is a sphere centered at O_i with the radius d_i. Some data points may be contained in the ranges of multiple partitions, such as M in the figure. Such a point only belongs to the partition whose reference point



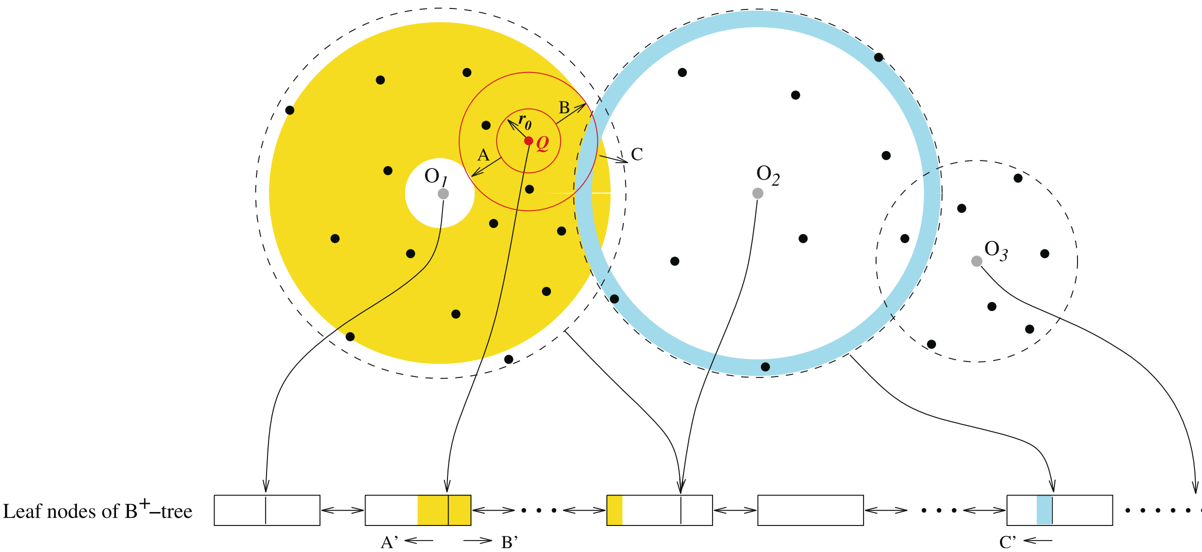
iDistance Techniques, Figure 1 KNN search of the iDistance

is closest to it. The closest reference point to M is O_3, so M belongs to P_3. Now the iDistance of a point p can be defined. Let P_i be the partition p belongs to. Then the iDistance of p

$$iDist(p) = dist(p, O_i) + i \cdot c\,,$$

where $dist(p_1, p_2)$ returns the distance between two points p_1 and p_2; i is the partition number and c is a constant used to stretch the data ranges. All points in partition P_i are mapped to the range $[i \cdot c, (i+1) \cdot c)$. c is set sufficiently large to avoid the overlap between the iDistance ranges of different partitions. Typically, it should be larger than the length of the diagonal in the hyper-rectangular data space. Given the definition of the iDistance, building the index is simple. The data points are stored in the leaf nodes of a B^+-tree using their iDistances as keys. Each partition corresponds to a continuous range of entries in the leaf nodes as shown in Fig. 1. Insertions and deletions are performed the same way as in a usual B^+-tree. The kNN search algorithm based on the iDistance index is described in the next paragraph.

Like the strategy of many other kNN search algorithms, the iDistance kNN search algorithm begins by searching a small "sphere", and incrementally enlarges the search sphere until the kNN are found. See the query point Q in Fig. 1. Let r denote the search radius. The search starts by a sphere region centered at Q with a initial radius r_0. Then r is iteratively increased by a small amount Δr until the kNN are found. At every iteration, the enlarged portion of the search sphere corresponds to an iDistance range (or two) for any intersected partition. All the points with their iDistances in the range(s) are retrieved in this iteration. During the whole process of enlarging the query sphere, the k nearest points to Q retrieved so far are maintained as candidates. Let p_{kth} be the kth nearest candidate point to Q. The algorithm terminates when $dist(p_{kth}, Q) \leq r$, which is the termination condition. At this moment, the k candidates are the actual kNN. For the example query Q in Fig. 1, its initial search sphere intersects partition P_1. In an intersected partition, there is a region that corresponds to the same iDistance range as the search sphere. This region is called the *mapped region* (of the query sphere), which is in the shape of an annulus around the reference point (the shaded region in P_1). All the points in the mapped region are retrieved. Note that all these points are stored continuously in the leaf nodes of the index, so the retrieval is a range query on the B^+-tree. As the current r is not greater than p_{kth}, it is increased by Δr and the search is continued. Figure 2 shows the enlarged search sphere and its mapped regions. Now the search sphere intersects P_2, too. The mapped region is a larger annulus in P_1 (the shaded region in P_1) and an annulus in P_2 (the shaded region in P_2). The enlarged portion of the annulus in P_1 consists of two smaller annulus regions, one expanding towards the center of P_1 (arrow A) and the other expanding outwards P_1 (arrow B). Each of the smaller annulus corresponds to a continuous range of points stored in the leaf nodes of the index and they adjoin the range retrieved last time. Therefore, the algorithm performs a backward range search (arrow A') and a forward range search (arrow B') to retrieve the new data points, starting from the boundaries of the last range retrieval. For P_2, the mapped region is just an annulus expanding towards the center of P_2 (arrow C) because Q lies outside of P_2. This corresponds to a range search backwards (arrow C') on the index. All the newly retrieved points are



iDistance Techniques, Figure 2 KNN search of the iDistance (continued)

compared with the maintained k candidates and always the k nearest ones to Q are maintained. After this, r continues to be increased iteratively until the termination condition is satisfied.

In the algorithm, r_0 and Δr are two other tunable parameters. r_0 can be set close to an estimation of the final search radius so that less iterations are needed to reach the final search radius, or r_0 can be simply set as 0. Δr is a small value so that not much unnecessary data points are retrieved. It can be set as 1% of an estimation of the final search radius. An implementation of the iDistance in C by Rui Zhang is available at http://www.csse.unimelb.edu.au/~rui/code.htm.

Key Applications

The iDistance technique is mainly designed for processing kNN queries in high-dimensional spaces. Typical applications are similarity search on multimedia data and data mining. Multimedia data such as images or videos are usually represented by features extracted from them including color, shape, texture, etc [2]. The number of features is large. Therefore, retrieving similar objects translates to kNN queries in high-dimensional spaces. In data mining or data warehouse, data records have large number of attributes. Finding similar objects also requires kNN search in high-dimensional spaces. In general, similarity search on any objects that can be represented by multidimensional vectors can take advantage of this technique.

Future Directions

The mapping function of the iDistance index relies on a fixed set of reference points. If the data distribution changes much, the kNN search performance may deteriorate. Then the iDistance index has to be rebuilt in order to keep the high performance. One open problem is the possibility to find a dynamic mapping function that can adapt to the change of data distributions automatically, so that the high performance is always kept without rebuilding the index.

Cross References

- ► Distance Metrics
- ► Indexing, High Dimensional
- ► Indexing of Moving Objects, B^x-Tree
- ► Nearest Neighbor Query
- ► Nearest Neighbors Problem

Recommended Reading

Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is nearest neighbors meaningful? In: Proceedings of International Conference on Database Theory (1999)

Faloutsos, C., Equitz, W., Flickner, M., Niblack, W., Petkovic, D., Barber, R.: Efficient and Effective Querying by Image Content. J. Intell. Inf. Syst. **3**(3), 231–262 (1994)

Jagadish, H.V., Ooi, B.C., Yu, C., Tan, K.-L., Zhang, R.: iDistance: An Adaptive B+-tree Based Indexing Method for Nearest Neighbor Search. ACM Trans. Data Base Syst. **30**(2), 364–397 (2005)

Weber, R., Schek, H., Blott, S.: A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In: Proceedings of International Conference on Very Large Data Bases (1998)

Yu, C., Ooi, B.C., Tan, K.-L., Jagadish, H.: Indexing the distance: an efficient method to KNN processing. In: Proceedings of International Conference on Very Large Data Bases (2001)

Image

► Oracle Spatial, Raster Data

Image Acquisition

► Photogrammetric Sensors

Image Analysis

► Data Acquisition, Automation

Image Compression

LEI SHA
Department of Physiology and Biophysics,
Mayo Clinic, Rochester, MN, USA

Synonyms

Data compression; Lossless image compression; Lossy image compression

Definition

Image compression is the process of encoding digital image information using fewer bits than an unencoded representation would use through use of specific encoding schemes.

Historical Background

Uncompressed image data is large in file size. To store or transmit the uncompressed image requires considerable storage capacity and transmission bandwidth. For example, for a uncompressed color picture file of 6,000 × 4,800 pixel (good for a mere 10 × 8 in. print at 600 dpi), number of bytes required to store it is 85 MB. Despite rapid progress in mass-storage density, processor speeds, and digital communication system performance, demand for image storage capacity and image transmission bandwidth constantly outstrip the technical progress. Image compression is therefore essential for image storage and transmission in most cases. In geographic information systems (GIS), images are generally very large. A simple uncompressed world map (3 × 2.5 ft. in size, for example) will require more than 1 GB storage space. To retrieve the simple world map remotely over Internet with an Internet transmission speed of 1 Mbps, it would take more than 2 h. GIS images are generally much larger than a simple world map. Therefore, how to compress and store GIS images (as well as fast access/retrieval) is a hot research topic.

Scientific Fundamentals

Principle of Image Compression

Images can be compressed, because images have some degree of redundancy. For example, one of the common characteristics of most images is that the neighboring pixels are correlated and, therefore, the information is redundant among the neighboring pixels. In still images, there are two type of redundancy: spatial redundancy and spectral redundancy.

Spatial redundancy refers to the correlation between neighboring pixels. This is the redundancy due to patterning, or self–similarity within an image. Spectral redundancy is the redundancy occurring on the correlation between different color planes or spectral bands. The objective of general image compression is to reduce redundancy of the image data in order to be able to store or transmit data in an efficient form.

There are two classes of image compression methods to reduce the redundancy: lossless image compression and lossy image compression. In lossless compression, the reconstructed image from the compression is identical to the original image. However, lossless compression can only achieve a modest amount of compression because the the redundant information is retained. Lossy image compression can achieve a high compression rate but an image reconstructed following lossy compression is degraded relative to the original, because the redundant information is completely discarded.

Lossless Image Compression

Lossless image compression uses a class of data compression algorithms that allows the exact original data to be reconstructed from the compressed data. Lossless compression is used when it is important that the original and the decompressed data be identical, or when no assumption can be made on whether a certain deviation is uncritical. Some image file formats, notably PNG (Portable network graphics), use only lossless compression, while others like TIFF (Tagged image file format) and MNG (Multiple-image Network Graphics) may use either lossless or lossy methods.

Most lossless compression programs use two different kinds of algorithms: one that generates a statistical model for the input data, and another that maps the input data to bit strings. The two type of lossless compression algorithmsare statistical modeling algorithms (for text or text-like binary data) and encoding algorithms to produce bit sequences. Statistical modeling algorithms include the Burrows–Wheeler transform (block sorting preprocessing that makes compression more efficient), LZ77 (used by

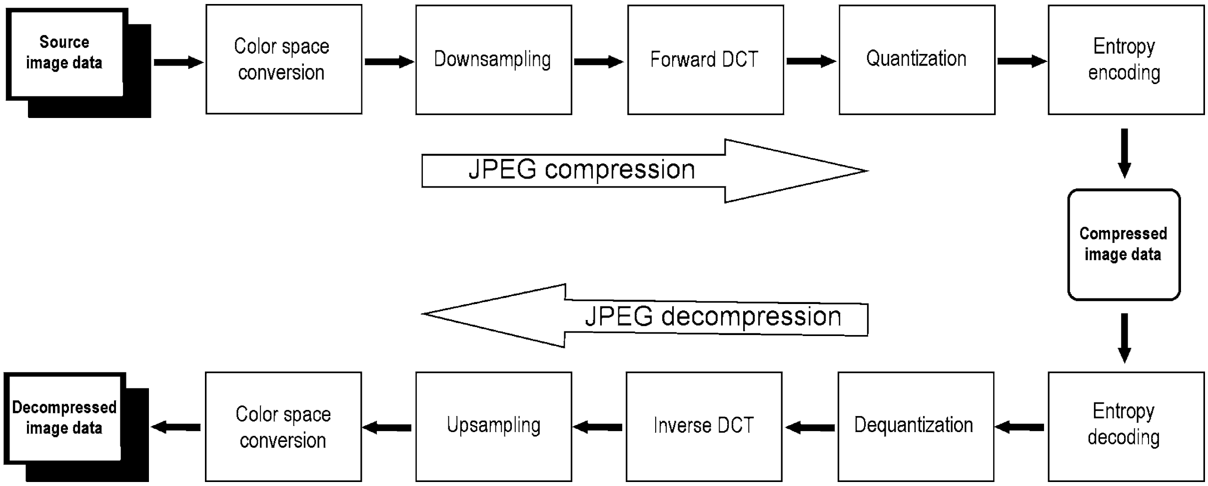


Image Compression, Figure 1 Compression and decompression flow in a popular lossy image compression. DCF (Discrete Cosine Transform)

DEFLATE) and LZW. Encoding algorithms include Huffman coding (also used by DEFLATE) and arithmetic coding.

Lossless data compression algorithms cannot guarantee compression of all input data sets. In other words, for any lossless data compression algorithm there will be an input data set that does not get smaller when processed by the algorithm.

Here are some popular lossless image compression format: adaptive binary optimization (ABO), GIF (Graphics Interchange Format, after lossy reduction of color depth), PNG, JPEG-LS (can be lossless/near-lossless), JPEG 2000 (includes lossless compression method), and TIFF (using Lempel-Ziv-Welch lossless compression).

Lossy Image Compression

A lossy compression method is one where compressing data and then decompressing it retrieves data that may well be different from the original, but is "close enough" to be useful in some way.

There are basically four lossy image compression methods:

- Transform coding: transform coding uses a transformation that exploits peculiar characteristics of the signal, increasing the performance of a scheme such as Huffman or arithmetic coding.
- Wavelet compression: wavelet compression is based on wavelet functions. It also adopts transform coding. The basic idea of this coding scheme is to process data at different scales of resolution. Wavelet compression has a very good scalability for macrostructure and microstructure as the compression is achieved using two versions of the picture as a different scaling of the same prototype function called the mother wavelet or wavelet basis.
- Vector quantization: this method is better known as the dictionary method. A dictionary is a collection of a small number of statistically relevant patterns. Every image is encoded by dividing it into blocks and assigning to each block the index of the closest codeword in the dictionary.
- Fractal compression: the basic idea of fractal compression is "self-vector-quantization," where an image block is encoded by applying a simple transformation to one of the blocks previously encoded. Algorithms based on fractals have very good performance and high compression ratios (32:1 is not unusual), but their use can be limited by the extensive computation required.

Some popular lossy image compression formats are: compression via fractal compression, JPEG, JPEG2000, JPEG's successor format that uses wavelets, wavelet compression, Cartesian perceptual compression, DjVu, and ICER (Incremental cost-effectiveness ratio).

Figure 1 is an example of compression and decompression flow in a popular lossy image compression (JPEG).

The advantage of lossy methods over lossless methods is that in some cases a lossy method can produce a much smaller compressed file than any known lossless method, while still meeting the requirements of the application. Lossy methods are most often used for compressing images. Lossily compressed still images are often compressed to a tenth of their original size.

When a user acquires a lossily compressed file, (for example, to reduce download time) the retrieved file can be quite different to the original at the bit level while being indistinguishable to the human eye for most practical purposes. Many lossy image compression methods focus on the idiosyncrasies of the human anatomy, taking into account, for example, that the human eye can see only certain frequencies of light. Although lossy image compression can achieve a very compact compression, high image compression can cause some artifacts.

Four types of artifacts may occur: blocking, blurring, ringing effect, and texture deviation. Between lossless image compressions and the varying degrees of lossy image compressions, there has to be some balance/trade-off for any particular mission. When considering using an image compression method, scalability of the compressed image using the image compression method is also very important. Especially, in the GIS field, scalability is important for viewing an image with different views, which provide variable quality access to databases.

Key Applications

In the GIS field, both lossless and lossy image compression methods are used depending on the differing demands of the various projects. JBIG(Joint Bi-level Image experts Group), for example, is a format using a lossless compression of a bilevel image. It can be used for coding grey scale and color GIS images with limited numbers of bits per pixel. It can offer between 20 and 80% improvement in compression (about 20 to 1 over the original uncompressed digital bit map). Formats based on JBIG can be used for large GIS binary images in digital spatial libraries.

However, in some projects, a high image compression rate and more information are needed. High resolution satellite images with more geometry and information content, for example, require a compression rate well above a factor of 10, and lossy image compression methods become necessary. In this case, JPEG/JPEG2000 or other image compression methods based on wavelet compression are widely used.

With many researchers' contributions to the spatial image compression technique, the spatial image compression field is rapidly moving forward. The following are a few new developments on spatial image compression research. Eugene Ageenko and Pasi Franti [9] proposed a compression method based on JBIG aimed at compression of large binary images in digital spatial libraries. The simulation results of their image compression method showed that their proposed method allows dense tiling of large images down to 50×50 pixels versus the 350×350 pixels that is possible with JBIG without sacrificing the compression performance. It will allow partial decompression of large images far more efficiently than if JBIG was applied. For clusters larger than 200×200 pixels, the method improves JBIG by about 20%. Renato Pajarola and Peter Widmayer [6,8] developed an image compression method for spatial search with the consideration of efficiency on decompression and retrieval of spatial images. The compression algorithm they proposed is oriented toward spatial clustering and permits decompression from local information alone. They demonstrated experimentally that the Hibert compression ratio typically is competitive with well known compression algorithms such as lossless JPEG or CALIC (Context-based Adaptive Lossless Image Coding). They implemented an experimental image database that provides spatial access to compressed images and can be used as a texture server to a real-time terrain visualization system. Hassan Ghassemian [11] developed an onboard satellite image compression method by object-feature extraction. Ghassemian stated that "recent developments in sensor technology make possible Earth observational remote sensing systems with high spectral resolution and data dimensionality. As a result, the flow of data from satellite-borne sensors to earth stations is likely to increase to an enormous rate." Therefore, "a new onboard unsupervised feature extraction method that reduces the complexity and costs associated with the analysis of multispectral images and the data transmission, storage, archival and distribution"was investigated. He developed an algorithm to reduce data redundancy by an unsupervised object-feature extraction process. His results showed an average compression of more than 25, the classification performance wasimproved for all classes, and the CPU time required for classification reduced by a factor of more than 25. Jiri Komzak and Pavel Slavik [5] in their paper described a general adaptive tool usable for compression and decompression of different types of data in GIS. Their approach makes it possible to easily configure compressor and decompressor and use different compression methods. There is also a very important possibility using their approach of optionally turning on and off lossy transitions during the compression process, and in such a way as to change the data loss even for each symbol. Jochen Schiewe [10] in his paper, and Ryuji Matsuoka and colleagues [7] in their paper took time comparing the effects of lossy data compression techniques on the geometry and information content of satellite imagery. They compared different image compression methods and outlined the advantages and disadvantages of a few popular methods. For example, Matsuoka concluded that, on comparison of lossy JPEG compression and lossy JPEG 2000 compression in performance, it was confirmed that lossy JPEG 2000 compression is superior to lossy JPEG compression in color features. However, lossy JPEG 2000 compression does not necessarily provide an image of good quality in texture features. Their research provided very useful information for GIS designers to use in picking image compression formats in their system.

As in the digital image era, the development of image compression techniques in GIS is progressing at a rapid pace.

Future Directions

The development of image compression technique in GIS is toward more efficient algorithms with high compression ratios, low magnitudes of the error introduced by the encoding (lossy image compression), and fast and scalable coding and retrieval.

Cross References

► Map Generalization

Recommended Reading

Wikipedia. Image compression. http://en.wikipedia.org/wiki/Image_compression

Saha, S.: Image compression—from DCT to wavelets: a review. http://www.acm.org/crossroads/xrds6-3/sahaimgcoding.html

Motta, G., Rizzo, F., Storer, J. A.: Digital image compression. In: Ralston, A., Reilly, E.D., Hemmendinger, D. (eds.) The Encyclopedia of Computer Science, 4th edn. Grove Dictionaries (2000)

GIS Standards. http://fortboise.org/maps/GISstandards.html

Komzak, J., Slavik, P.: Architecture of system for configurable GIS data compression. http://kmi.open.ac.uk/people/jiri/publ/wscg2002.pdf

Pajarola, R., Widmayer, P.: Spatial indexing into compressed raster images: how to answer range queries without decompression. http://www.ifi.unizh.ch/vmml/admin/upload/MMDBMS96.pdf

Matsuoka, R., Sone, M., Fukue, K., Cho, K., Shimoda, H.: Quantitative analysis of image quality of lossy compression images. http://www.isprs.org/istanbul2004/comm3/papers/348.pdf

Pajarola, R., Widmayer, P.: An image compression method for spatial search. IEEE Trans. Image Process. **9**, (2000)

Ageenko, E.I., Frnti, P.: Compression of large binary images in digital spatial libraries. Comput. Graphics **24**, 91–98 (2000)

Schiewe, J.: Effect of lossy data compression techniques of geometry and information content of satellite imagery. IAPRS. **32**, 540–544 (1998)

Hassan Ghassemain. On-board satellite image compressiaon by object-feature wxtraction. International Atchieves of Photogrammertry Remote Sensing and Spatial Information Sciences.**35**, 820–825 (2004)

Image Mining, Spatial

SUDHANSHU SEKHAR PANDA
Department of Science, Engineering, and Technology, Gainesville State College, Gainesville, GA, USA

Synonyms

Visual data mining; Image pattern; Object recognition; Association rules: image indexing and retrieval; Feature extraction

Definition

Image mining is synonymous to data mining concept. It is important to first understand the data mining concept prior to image mining. Data mining is a set of techniques used in an automated approach to exhaustively explore and establish relationships in very large datasets. It is the process of analyzing large sets of domain-specific data and subsequently extracting information and knowledge in a form of new relationships, patterns, or clusters for the decision-making process [1]. Data mining applications are of three-level application architecture. These layers include applications, approaches, and algorithms and models [2]. The approaches of data mining are association, sequence-based analysis, clustering, estimation, classification, etc. Algorithms and models are then developed based on the dataset type to perform the data mining. At that point, the ill-data points are extracted from the dataset. Similarly, image mining is a set of tools and techniques to explore images in an automated approach to extract semantically meaningful information (knowledge) from them. A single image is a collection of a set of data. Therefore, image mining techniques are far more complicated than the data mining techniques, but data mining tools and techniques could be replicated for the purpose of image mining. However, there should not be any misnomer that image mining is just a simple extension of data mining applications or just a pattern recognition process [3].

Historical Background

Images are abundant in the present multimedia age. There has been tremendous growth in significantly large and detailed image databases due to the advances in image acquisition and storage technology [4]. The World Wide Web is also regarded as the largest global image repository. Every day, a large number of image data are being generated in the form of satellite images, aerial images, medical images, and digital photographs. These images, if analyzed with proper tools and techniques, can reveal useful information to the users. However, there is general agreement that sufficient tools are not available for analysis of images [3]. Effective identification of features in the images and their proper extraction are some of the important problems associated with image analysis. One of the difficult tasks is to know the image domain and obtain a priori knowledge of what information is required from the image. Therefore, complete automation of an image mining process cannot be performed. Image mining deals with the extraction of implicit knowledge, image data relationship, or other patterns not explicitly stored in the image databases. It is an interdisciplinary endeavor that essential-

I

ly draws upon expertise in computer vision, image processing, image retrieval, data mining, machine learning, database, and artificial intelligence [5].

Scientific Fundamentals

Issues in Image Mining

Image mining, by definition deals with the extraction of image patterns from a group of images. Image mining is not exactly similar to low-level computer vision and image processing techniques. The computer vision and image processing techniques are applied to understand and/or extract specific features from a single image where the objective of image mining is to discover image patterns in a set of collected images [3]. As mentioned earlier, image mining is not just another data mining technique.

Difference Between Image Mining and Data Mining

In a relational database, the data are well defined. While analyzing a relational database of annual precipitation and runoff, a value of 15 cm rainfall in May, 2006 has its own absolute meaning. It suggests that there is 15 cm of rainfall in May, 2006. But in an image database (which deals with mostly the digital gray values), a gray value of 15 representing a moist soil in one image could be 25 in another image based on all other pixels in the image, i. e., if the second image is taken with a brighter light condition than the first one, all digital gray values will end up in the second image as higher numbers. Therefore, image mining deals with relative values, while data mining always deals with absolute values.

Image mining always deals with spatial issues as images represent spatial references. Satellite images and aerial photographs are representing spatial locations of earth. Similarly, when analyzing a tumor in an image of a body part, the tumor position is analyzed on a spatial basis, too, i. e., gauging how far is the tumor from the lungs, intestine, or brain center, etc. In case of a digital image of a person, the spatial position of different features should be analyzed. But in case of data mining, spatial consideration may not be required. Consequently, image miners try to overcome the spatial issue problem by extracting position-independent features from images before attempting to mine useful patterns from the images [3].

Often image mining deals with multiple interpretations of images of same location (Fig. 1). For example, temporal changes of the land use/land cover of an area need to be analyzed differently than the plain data mining or feature extraction techniques, to extract the change information in images. Therefore, traditional data mining techniques of associating a pattern to a class interpretation may not work with image mining. A new set of algorithms is needed to cater to discover useful patterns from images [3].

Image Mining Techniques

Image mining may use data from a image databases (Fig. 1). Basically, images are stored with some description against a particular image. Again images are nothing but some intensity values, which figure the image in terms of color, shape, texture, etc. The mining task is based on using such information contained in the images [6]. The following are some of the image mining techniques frequently used by image miners:

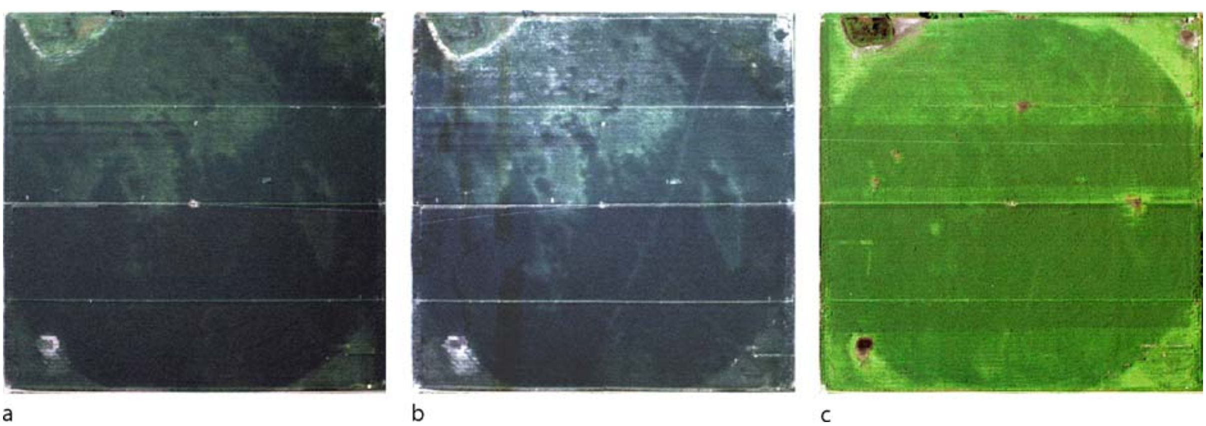


Image Mining, Spatial, Figure 1 Set of aerial images (part of image database) of a quarter size crop field used for crop yield analysis using image mining techniques. **a** NW29 quarter aerial image in August 1997. **b** W29 quarter aerial image in August1997 (few days apart). **c** W29 quarter aerial image in August 1998 (similar time of the month as first one). Note: The images quality and patterns in images change as a factor of difference in image acquisition set up and other environmental factor

1. Image acquisition
2. Image enhancement and restoration
3. Object recognition
4. Image indexing and retrieval
5. Image segmentation and feature extraction
6. Representation and description
7. Association rules mining

Among these tasks, image indexing and retrieval and image segmentation and extraction are two of the principal tasks in image mining. They are categorized as content-based techniques [7]. Some of the other image retrieval techniques are categorized under description based retrieval techniques.

1) ***Image acquisition***: In image mining, the understanding of the image acquisition setup is very much essential. As described earlier, similar spatial images vary in look with different image acquisition setups, such as lighting intensity, use of sensor type, and radiometric factor (in case of satellite and aerial images), etc. While comparing a set of similar spatial images, it is necessary to bring them into similar image acquisition setup. Color calibration is a technique that could be employed for this. For satellite images, radiometric correction techniques can be used. ENVI software (ITT Visual Information Solutions, Boulder, CO) has a good technique for radiometric corrections of satellite images.
2) ***Image enhancement and restoration***: Image enhancement and restoration is one of the simplest image processing techniques [8]. This technique is used to highlight only the important area of interest in an image and ignoring irrelevant areas. Contrast enhancement is an example of image enhancement and restoration. Log transformations, power-law transformations, piecewise-linear transformation, histogram equalization, image subtraction, image averaging, smoothing by different spatial filters, sharpening by different spatial filters, Fourier transformation, noise reduction using different filtering techniques, geometric transformations, etc., are some of the algorithms used for image enhancement and restoration. Books [8,9,10] on image processing will provide with the algorithms and working procedure for all these techniques.
3) ***Object recognition***: Object recognition is one of the major tasks in image mining. With the help of a known object models, an object recognition system finds objects in the real world from an image. Automatic machine learning and meaningful information extraction can be comprehended by training (supervised training) the machine with some known objects [3]. Machine learning algorithms could be accessed from several neural networks books [11]. The object recognition model is initially provided with a set of objects and labels (those are expected to be found in the tested image) to become trained and the model for object recognition assigns correct labels to regions, or a set of regions, in the image. Models of known objects and labels are usually provided by human (teacher for the model).
   In general, an object recognition module consists of four components, such as 1) model database, 2) feature detector, 3) hypothesizer, and 4) hypothesis verifier. The model database contains all the models known to the system that became trained. The models also contain several features or labels that describe the objects in the image being tested. The detected image primitive features in the gray scale (pixel) level help the hypothesizer to assign likelihood to the objects in the image. Finally, the verifier uses the models to verify the hypothesis, refine the object likelihood, and label the features in the tested image [3,5]. Thus the correct objects are recognized by the machine learning model. Bonet [12] designed a system that processes an image into a set of "characteristic maps" to locate a particular known object in an image or set of images.
4) ***Image indexing and retrieval***: Image mining is incomplete without image retrieval process. Images can be retrieved based on features, such as color, texture, shape, size, or the spatial location of image elements; logical features like objects of a given type or individual objects or persons; or abstract attributes or features. Three query schemas for image retrieval are described by Rick Kazman and Kominek [13]. They include query by associate attributes, query by description, and query by image content. In recent times, the content-based image retrieval techniques are used most [14]. Content-based image retrieval is based on three processes, such as, visual information extraction, image indexing and retrieval system application [15]. IBM's QBIC system [16] is perhaps the best known of all image content retrieval systems in commercial application. The system offers image retrieval by individual parameter or any combination of color, texture, shape, or text keyword along with R-tree indexing feature to improve search efficiency [3]. There are many other software commercially available for image indexing and retrieval. A simple web search with "image indexing software" can provide insight to them.
5) ***Image segmentation and feature extraction***: Image segmentation and feature extraction are closely similar to image indexing and retrieval process. Image segmentation can be performed on a single image or a batch of images based on batch processing. Segmentation procedures partition an image into constituent parts or objects. They are mostly feature-based, i. e., by analyz-

ing image features such as color, texture, shape, and size, etc. Intelligently segmenting an image by content is an important way to mine valuable information from large image collection. Several image segmentation techniques are used now a day. New techniques are evolving with the advancement in computer programming. Some of the image segmentation techniques are ISODATA, K-means, Fuzzy C-means, region growing, self organizing map (SOM) neural classification, Bayesian classifier based hierarchical classification, WARD-minimum variance method, maximum likelihood, etc. They are mostly classified into two main categories, such as supervised and unsupervised. Unsupervised segmentation techniques are widely pursued because it does not need any a priori knowledge regarding the image under segmentation. However, the better the segmentation algorithm, there is a good probability of correct feature extraction. Image segmentation in multimedia is a priority now. They are although similar to the normal segmentation procedures used in remote sensing image analysis but work differently. Zaiane and Han [4] have developed MM-Classifier, a classification module embedded in the MultiMedia Miner which classifies multimedia data, including images. Wang and Li [17] have proposed a new classifier technique, IBCOW (image-based classification of objectionable websites) to classify websites based on objectionable or benign image content.

Artificial intelligence has a big role in image segmentation now. Artificial neural network (ANN) and self-organizing maps are being used to accurately segment images for several applications [18].

6) ***Representation and description***: In image mining, these steps always follow the output of segmentation stage. The output of segmentation is usually the raw pixel data constituting either boundary of a region or all the points of a region. Converting these output data to a form suitable for computer processing is essential [8]. The decision taken to represent the classified pixels as boundary pixels or pixels of a region is the representation stage. Finally, description of region based on real world completes the image classification process in image mining.
7) ***Association rule mining***: There is a two steps approach in a typical association rule mining algorithm. The first step finds all large item sets that meet the minimum support constraint while the second step generates rules from all the large item sets that satisfy the minimum confidence constraint [3]. Association rule mining generated rules works well in image mining than some user-defined thresholds [3]. Association rule mining is used in data mining to uncover interesting trends, patterns, and rules in large datasets. Association rule mining has found its application in image mining consisting of large image databases [19,20]. The first approach in association rule mining in image mining is to mine from large collections of images alone and the second approach is to mine from the combined collections of images and associated alphanumeric data [20]. Simon Fraser University has developed a prototype called Multimedia Miner, and one of its major modules is called MM-Associator, which uses a three-dimensional visualization to explicitly display the associations [4].

Key Applications

Image mining is a recent phenomenon. It is still in its infancy. It has lot of scope in multimedia research.

World Wide Web

Web image mining is a very new concept. With the advent of Google, Yahoo, AltaVista, and other web search engines, it is possible to search articles and images with text-based search method. But searching images from World Wide Web using a visual approach is a challenge. However, web image mining techniques are progressive steps towards that [21].

Multimedia

As discussed in the previous text, multimedia has the largest application of image mining. They deal with huge databases of images, animations, audios, videos, and other text databases. Retrieving information from these databases is getting easier with the use of several image mining software [22]. Several other software programs developed for this purpose are mentioned in the main text.

Healthcare

Modern health care generates huge amounts of digital data, including an overabundance of images. They include SPECT images, DNA micro-array data, ECG signals, clinical measurements, such as blood pressure and cholesterol levels, and the description of the diagnosis given by the physician interpretation of all these data. With the advent of image mining techniques, most of their processes are becoming automated.

Engineering and Construction Industry

Architectural designs using AutoCAD generates huge amount of digital data. Visual data generated in engineering and construction industries are 3D-models, building component databases, images of the environment, text descriptions and drawings from the initial and evolved design requirements, financial and personnel data, etc. They could profit from image mining.

Geotechnology

It is obvious that the geotechnology industry representing GIS, remote sensing, and GPS application deals with highest amount of image databases. These databases are generated in any field of application, such as agriculture, forestry, environmental science, geosciences, to name a few. Image mining is a premium tool in geotechnology field. VisiMine (Insightful Corporation, Seattle, WA) is an image mining software (a search engine) used for analyzing image databases. It is designed for satellite imagery and aerial photos analysis. Visimine provides a comprehensive workbench for image information mining by integrating state-of-the-art statistics, data mining, and image processing to extract information and locate images from potentially huge databases.

Police and Security

The organizations dealing with security are swamped with image databases. Image mining techniques are having a major use in their activities.

Future Directions

As image mining is still in its infancy, there is a huge scope for its development. Future research should emphasize on these few factors: 1) design powerful query language (a la GIS query expression) to access image data from image database, 2) explore new visual discovery algorithm to locate unique characteristics present in image data, and 3) devise automated image mining techniques based on content based image retrieval techniques.

Cross References

► Co-location Patterns, Algorithms
► Gaussian Process Models in Spatial Data Mining

Recommended Reading

1. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, San Francisco, CA (2001)
2. Moxon, B.: Defining Data Mining: The Hows and Whys of Data Mining, and How it Differs from Other Analytical Techniques. DBMS Data Warehouse Supplement, Miller Freeman, Inc., San Francisco, CA (1996)
3. Zhang, J., Hsu, W., Lee, M.L.: An information-driven framework for image mining. In: Proceedings of 12th International Conference on Database and Expert Systems Applications (DEXA), Munich, Germany (2001)
4. Zaiane, O.R., Han, J.W.: Mining multimedia data. CASCON'98: Meeting of Minds, pp 83–96, Toronto, Canada, November 1998
5. Burl, M.C.: Mining for image content. In Systemics, Cybernetics, and Informatics / Information Systems: Analysis and Synthesis, Orlando, FL (July 1999)
6. Jain, L. C., Ghosh, A.:Evolutionary Computation in Data Mining. Springer, New York, NY (2005)
7. Rui, Y., Huang, T., Chang, S.: Image retrieval: current techniques, promising directions and open issues. J. Vis. Commun. Image Represent. **10**(4): 39–62 (1999)
8. Gonzalez, R. C., Woods, R. E.: Digital Image Processing. 2nd edn. Pearson Education, New Delhi, India (2002)
9. Jain, A.K.: Fundamentals of Digital Image Processing. Prentice Hall, New York, NY (1998)
10. Pitas, I.: Digital Image Processing Algorithms. Prentice Hall, New York, NY (1993)
11. Haykin S.: Neural Networks a Comprehensive Foundation. Prentice Hall Inc., New York, NY (1999)
12. Bonet, J.S.D.: Image Reprocessing for Rapid Selection in "Pay attention mode". MIT Press, Boston, MA (2000)
13. Kazman, R., Kominek, J.: Information organization in multimedia resources. Proceedings of the 11th annual international conference on Systems documentation, pp149–162 (1993)
14. Foschi, P. G., Kollipakkam, D., Liu, H., Mandvikar, A.: Feature extraction for image mining (2007) www.public.asu.edu/~huanliu/papers/mis02.pdf. Accessed on 01.10.07
15. Rui, Y., Huang, T.S.: Image retrieval: past, present and future. Invited paper in: International Symposium on Multimedia Information Processing, Taipei, Taiwan, December 11–13, 1997
16. Niblack, W., Barber, R.: The QBIC project: querying images by content using color, texture and shape. In: Proc. SPIE Storage and Retrieval for Image and Video Databases, February 1994
17. Wang, J.Z., Li, J.:System for screening objectionable images using daubechies' wavelets and color histograms. Interactive Distributed Multimedia Systems and Telecommunication Services, Proceedings of the Fourth European Workshop (IDMS'97) (1997)
18. Panda, S.: Data mining application in production management of crop. Ph.D. Dissertation, North Dakota State University, Fargo, North Dakota, USA (2003)
19. Megalooikonomou, V., Davataikos, C., Herskovits, E. H.: Mining lesion-deficit associations in a brain image database. KDD, San Diego, CA, USA (1999)
20. Ordonez, C., Omiecinski, E.: Discovering association rules based on image content. In: Proceedings of the IEEE Advances in Digital Libraries Conference (ADL'99) (1999)
21. Yanai, K.: Web image mining toward generic image recognition. In: Proceedings of the Twelfth International World Wide Web Conference (2003)
22. Missaoui, R., Sarifuddin, M., Vaillancourt, J.: Similarity measures for efficient content-based image retrieval. In: IEE Proceedings – Visual Image Signal Process. 152: 6, December 2005

Image Pair

► Photogrammetric Methods

Image Pattern

► Image Mining, Spatial

Image Station

► Intergraph: Real Time Operational Geospatial Applications

Image Triplet

► Photogrammetric Methods

Imagery Conflation

► Conflation of Geospatial Data

Immediate Response Zone

► Emergency Evacuations, Transportation Networks

Imprecision and Spatial Uncertainty

MICHAEL F. GOODCHILD
Department of Geography, University of California
Santa Barbara, Santa Barbara, CA, USA

Synonyms

Quality, spatial data; Accuracy, spatial; Accuracy, map; Error propagation

Definition

Spatial uncertainty is defined as the difference between the contents of a spatial database and the corresponding phenomena in the real world. Because all contents of spatial databases are representations of the real world, it is inevitable that differences will exist between them and the real phenomena that they purport to represent. Spatial databases are compiled by processes that include approximation, measurement error, and generalization through the omission of detail. Many spatial databases are based on definitions of terms, classes, and values that are vague, such that two observers may interpret them in different ways. All of these effects fall under the general term of spatial uncertainty, since they leave the user of a spatial database uncertain about what will be found in the real world. Numerous other terms are partially synonymous with spatial uncertainty. Data quality is often used in the context of metadata, and describes the measures and assessments that are intended by data producers to characterize known uncertainties. Vagueness, imprecision, and inaccuracy all imply specific conceptual frameworks, ranging from fuzzy and rough sets to traditional theories of scientific measurement error, and whether or not it is implied that some true value exists in the real world that can be compared to the value stored in the database.

Historical Background

Very early interest in these topics can be found in the literature of stochastic geometry (Kendall, 1961), which applies concepts of probability theory to geometric structures. An early paper by Frolov and Maling (1969) analyzed the uncertainties present in finite-resolution raster representations, and derived confidence limits on measures such as area, motivated in part by the common practice of estimating measures of irregular patches by counting grid cells. Maling's analysis established connections between the spatial resolution of the overlaid raster of cells and confidence limits on area estimates. Maling's book (Maling, 1989) was a seminal venture into the application of statistical methods to maps, and helped to stimulate interest in the topic of spatial uncertainty. The growth of geographic information systems (GIS) provided the final impetus, and led to the first research initiative of the new US National Center for Geographic Information and Analysis in 1988, on the topic of accuracy in spatial databases (Goodchild and Gopal, 1989).

The notion that spatial databases could be treated through the application of classical theories of measurement error soon proved too limiting, however. The definitions of types that are used in the compilation of maps of soil class, vegetation cover class, or land use are clearly open to interpretation, and such maps must be regarded as to some degree subjective and outside the normal bounds of scientific replicability. Concepts of fuzzy and rough sets were explored by researchers interested in these issues (Fisher and Unwin, 2005). While the definition of a given class may be vague, it is nevertheless helpful to think about degrees of membership in the class. For example, researchers interested in developing plain-language interfaces to GIS found that prepositions such as "near" had vague meanings that could be represented more formally through membership functions. This approach resonated well with the move in the early 1990s to introduce theories of linguistics and cognition into GIS research.

By the end of the 1990s the literature on spatial uncertainty had grown to include several distinct theoretical frameworks, including geostatistics, fuzzy sets, rough sets, and spatial statistics. Zhang and Goodchild (2002) published a synthesis, framed within the fundamental dichotomy between discrete objects and continuous fields that under-lies much of GIScience. Research continues, particularly on such topics as spatial uncertainty in digital terrain data.

Scientific Fundamentals

In the classical theory of measurement, an observed value z' is distorted from its true value z by a series of random effects. If these effects are additive, the distortion $\delta z = z' - z$ is expected to follow a Gaussian distribution, and each observed measurement is interpreted as a sample drawn from that distribution. The mean of the distribution is termed the bias or systematic error, and the root mean square of δz is termed the standard error. The standard deviation of δz with respect to its own mean is often termed precision, and a biased measurement device is thus said to be possibly precise but not accurate. However, precision can also refer to the number of numerical digits used to report a measurement, and imprecision is used in several ways in the literature on spatial uncertainty.

This analysis extends readily to measurement of position in two or three dimensions, and thus to measurements made by such technologies as the Global Positioning System (GPS), where the multivariate Gaussian distribution is widely used to characterize positional uncertainty. Measurement errors in the two horizontal dimensions are commonly found to have equal variance, but errors in the vertical dimension typically have very different variance; and measurement errors in all three dimensions are commonly found to be uncorrelated.

This classical theory has been developed extensively within the discipline of surveying, under the rubric of adjustment theory, in order to understand the effects that errors in raw measurements may have on the inferred locations of items of interest. For example, errors in the measurement of bearing, elevation, and range will translate into errors in the inferred positions of the objects of the survey. Complications arise when closed loops are surveyed in the interests of reducing errors, which must then be allocated around the loop in a process known as adjustment. This body of theory has not had much influence on spatial databases, however, outside of the domain of traditional surveying.

Any spatial database will consist of large numbers of measurements. For example, a remotely sensed image may contain millions of pixels, each containing several measurements of surface reflectance. Although measurements made by simple devices such as thermometers can reasonably be assumed to have statistically independent errors, this is almost never true of data compiled across geographic space. Instead, strong and mostly positive correlations are observed between data values that are close together in space. These correlations may be induced by the production process, when many data values inherit the errors in a smaller number of nearby measurements through various forms of interpolation, or through the measurements themselves, which are distorted by effects that operate across areas of space. Such correlations are generally known as spatial dependence or spatial autocorrelation.

This tendency turns out to be quite useful. For example, consider a curved segment of a street, recorded in a spatial database as a sequence of coordinate pairs. Assume a measurement error of 10 m, not unreasonable in today's street centerline databases. If each point was independently disturbed by 10 m, the result would be impossibly and unacceptably erratic, and the segment's length as determined from the database would be severely overestimated. Instead, positive correlation between nearby errors ensures that the general shape of the street will be preserved, even though its position is disturbed. Similar arguments apply to the preservation of slopes in disturbed elevation models, and to many other examples of spatial data.

Several authors have drawn attention to an apparent paradox that follows from this argument. Consider a straight line, such as a straight segment of a street or property boundary, and suppose that the endpoints are disturbed by measurement error. If the disturbances are independent with known distributions, standard errors can be computed at any point along the line; and are found to be in general smaller away from the endpoints. If the disturbances have perfect positive correlation then standard errors are constant along the line; if they have identical and independent distributions then standard error is least at the midpoint where it is equal to 0.707 times the endpoint standard error; and if errors have perfect negative correlation then standard error will drop to zero at one intermediate point. Kyriakidis and Goodchild (2006) have generalized this problem to several other instances of linear interpolation. In practice, however, the straight line may itself be a fiction, and deviations of the truth from the straight line will tend to rise away from the endpoints, more than compensating for this effect.

Geostatistics (Goovaerts, 1997) provides a comprehensive theoretical framework for modeling such spatial autocorrelation of errors. Variances between nearby errors are expected to increase monotonically up to a distance known as the range, beyond which there is no further increase. The variance at this range is termed the sill, and corresponds to the absolute error of the database; however relative error is less over distances shorter than the range, and near zero over very short distances. Mathematical functions provide models of the monotonic increase of variance with distance.

Such models provide a convenient and powerful basis for exploring the effects of errors in applications such as terrain databases. Just as one might simulate the effects of error by adding independent samples from a Gaussian distribution to an observed value, so the effects of error in such databases can be simulated by adding realizations from random field models with suitable spatial covariances. In such cases, however, and because of the strong spatial dependences present in virtually all spatial data, it is the entire database that must be simulated in each realization of the random process, not its individual measurements; and samples from the stochastic process are entire maps, not simple measurements. Such simulations have proven very useful in visualizing the effects of spatially autocorrelated errors in spatial databases, and in exploring the propagation of such errors during GIS analysis. Several studies have demonstrated the use of geostatistical techniques such as conditional simulation to provide models of error in spatial databases.

Progress has been made in modeling the ways in which uncertainties propagate through GIS operations based on this theoretical framework. Although simple queries may refer only to a single point, and require knowledge only of that point's marginal distribution of uncertainty, other operations such as the measurement of area, distance, slope, or direction require knowledge of joint distributions and thus covariances. Heuvelink (1998) has developed a comprehensive framework for the propagation of uncertainty, using both analytic and numeric methods, including Taylor series approximations.

Such approaches are fundamentally limited by their insistence on the existence of a truth that is distorted by measurement. They fit well with applications in terrain modeling, and the positional accuracy of well-defined features such as roads, but poorly to applications involving classifications of soil, vegetation cover, or land use. But progress has been made in analyzing these latter types of database using the theoretical frameworks of fuzzy and rough sets. Briefly, such frameworks suppose that although the exact nature of a class A may remain unknown, it is still possible to measure membership $m(\text{A})$ in the class. Zhu et al. (1996) have shown how maps of membership can be useful in characterizing inherently vague phenomena, and Woodcock and Gopal (2000) have shown how such maps can be useful in managing forests. Fisher and Unwin (2005) have explored more advanced versions of these simple frameworks. Fundamentally, however, and despite the simplicity and intuitive appeal of these approaches, the question remains: if A cannot be defined, how is it possible to believe that $m(\text{A})$ can be measured? Moreover, it has proven difficult to represent the fundamental spatial dependence properties of spatial data within these frameworks, so while marginal properties can be analyzed with some success, the joint properties that underlie many forms of GIS analysis remain the preserve of statistical methods and of frameworks such as geostatistics and spatial statistics.

Key Applications

The literature on imprecision and spatial uncertainty now encompasses virtually all types of spatial data. As noted earlier, the literature on uncertainty in terrain data is voluminous. Several authors have demonstrated the use of representations of uncertainty in spatial decision support (e. g., Aerts, Goodchild, and Heuvelink, 2003), and have discussed the many sources of uncertainty in such applications. Interesting methods have been devised for visualizing uncertainty, including animation (Ehlschlaeger, Shortridge, and Goodchild, 1997). To date, however, the implementation of these methods in GIS software remains limited. Duckham (2002) and Heuvelink (2005) have described efforts to build error-aware systems, and data quality is now an important element of metadata. But the mainstream GIS products continue to report the results of calculations to far more decimal places than are justified by any assessment of accuracy, and to draw lines whose positions are uncertain using line widths that are in no way representative of that uncertainty. Indeed, GIS practice seems still to be largely driven by the belief that accuracy is a function of computation, not representation, and that the last uncertainties were removed from maps many decades ago.

Future Directions

Uncertainty has been described as the Achilles' Heel of GIS (Goodchild, 1998): the dark secret that once exposed, perhaps through the arguments of clever lawyers, will bring down the entire house of cards. While this sounds extreme, it is certainly true that the results of GIS analysis are often presented as far more accurate than they really are. As GIS moves more and more into the realm of prediction and forecasting, the dangers of failing to deal with uncertainty are likely to become more and more pressing. At the same time the accuracy of databases is steadily improving, as more accurate measurements become available. Nevertheless there is an enormous legacy of less accurate data that is sure to continue to find application for many years to come.

While much has been learned over the past two decades about the nature of spatial uncertainty, a large proportion of the literature remains comparatively inaccessible, due to the complexity of its mathematics. Some progress has been made in making the work more accessible, through visualization and through the comparatively straightforward methods of Monte Carlo simulation. In time, such approaches will result in greater awareness of what is possible, and greater adoption of these methods within the wider community.

Progress is also needed on the construction of suitable data models for error-sensitive spatial databases. The simple expedient of adding values representing uncertainty to the entire database in its metadata, or to individual objects as additional attributes, fails to capture all of the complexity of spatial uncertainty, particularly its essential spatial dependence. Goodchild (2004) has argued that this problem is profound, stemming from the complex structures of spatial dependence; that it presents fundamental and expensive barriers to any attempt to improve spatial databases through partial correction and update; and that it can only be addressed by a radical restructuring of spatial databases around the concept of measurement, in what he terms a measurement-based GIS. In practice, the almost universal use of absolute coordinates to define position in spatial databases ensures that any information about spatial dependence, and the processes used to compute or compile such positions, will have been lost at some point during the production process.

Cross References

► Statistical Descriptions of Spatial Patterns

Recommended Reading

1. Aerts, J.C.J.H., Goodchild, M.F., Heuvelink, G.B.M.: Accounting for spatial uncertainty in optimization with spatial decision support systems. Trans. GIS **7**(2), 211–230 (2003)
2. Duckham, M.: A user-oriented perspective of error-sensitive GIS development. Trans. GIS **6**(2), 179–194 (2002)
3. Ehlschlaeger, C.R., Shortridge, A.M., Goodchild, M.F.: Visualizing spatial data uncertainty using animation. Comput. Geosci. **23**(4), 387–395 (1997)
4. Fisher, P.F., Unwin, D.J. (eds.): Re-Presenting GIS. Wiley, Hoboken, NJ (2005)
5. Frolov, Y.S., Maling, D.H., The accuracy of area measurement by point counting techniques. Cartogr. J. **1**, 21–35 (1969)
6. Goodchild, M.F.: Uncertainty: the Achilles heel of GIS? Geo. Info. Syst. November, 50–52 (1998)
7. Goodchild, M.F.: A general framework for error analysis in measurement-based GIS. J. Geogr. Syst. **6**(4), 323–324 (2004)
8. Goodchild, M.F., Gopal, S.: Accuracy of Spatial Databases. Taylor and Francis, New York (1989)
9. Goovaerts, P.: Geostatistics for Natural Resources Evaluation. Oxford, New York (1997)
10. Heuvelink, G.B.M.: Error Propagation in Environmental Modelling with GIS. Taylor and Francis, London (1998)
11. Heuvelink, G.B.M.: Handling spatial uncertainty in GIS: development of the data uncertainty engine. Instituto Geografico Portugues, Estoril (Portugal) (2005)
12. Kendall, M.G.: A Course in the Geometry of *n* Dimensions. Hafner, New York (1961)
13. Kyriakidis, P., Goodchild, M.F.: On the prediction error variance of three common spatial interpolation schemes. Int. J. Geogr. Inf. Sci. **20**(8), 823–856 (2006)
14. Maling, D.H., Measurement from Maps: Principles and Methods of Cartometry. Pergamon, New York (1989)
15. Woodcock, C.E., Gopal, S.: Fuzzy set theory and thematic maps: accuracy assessment and area estimation. Int. J. Geogr. Inf. Sci. **14**(2), 153–172 (2000)
16. Zhang, J.-X., Goodchild, M.F.: Uncertainty in Geographical Information. Taylor and Francis, New York (2002)
17. Zhu, A.X., Band, L.E., Dutton, B., Nimlos, T.: Automated soil inference under fuzzy logic. Eco. Mod. **90**(2), 123–145 (1996)

Incident Management System

► Emergency Evacuation Plan Maintenance

Index Lifetime

► Indexing Schemes for Multi-dimensional Moving Objects

Index, MVR-Tree

► Indexing Spatio-temporal Archives

Index, R-Tree

► Indexing Spatio-temporal Archives

Index Structures, Extensible

MARIOS HADJIELEFTHERIOU[1], ERIK HOEL[2],
VASSILIS J. TSOTRAS[3]
[1] AT&T Labs Inc., Florham Park, NJ, USA
[2] ESRI, Redlands, CA, USA
[3] Computer Science Department,
University of California Riverside, Riverside, CA, USA

Synonyms

Indexing framework, spatial/spatiotemporal; Indexing API, spatial/spatiotemporal; Library, software

Definition

SaIL (SpAtial Index Library) [15] is an extensible application programming framework that enables easy integration of spatial and spatio-temporal index structures into existing applications. SaIL focuses mainly on design issues and techniques for providing an application programming interface generic enough to support user defined data types, customizable spatial queries, and a broad range of spatial and spatio-temporal index structures, in a way that does not compromise functionality, extensibility and, primarily, ease of use.

Historical Background

A plethora of GIS and other applications are related with spatial, spatio-temporal and, generally, multi-dimensional data. Typically, such applications have to manage millions of objects with diverse spatial characteristics. Examples include mapping applications that have to visualize numerous layers and hundreds of thousands of features [8], astronomical applications that index millions of images [20], and traffic analysis and surveillance applications that track thousands of vehicles. The utility of spatial indexing techniques for such applications has been well recognized—complex spatial queries can be answered efficiently only with the use of spatial index structures (for instance nearest neighbor queries). Consequently, many indexing techniques aiming to solve disparate problems and optimized for diverse types of spatial queries have appeared lately in the literature [4,11]. As a result, each technique has specific advantages and disadvantages that make it suitable for different application domains and types of data. Therefore, the task of selecting an appropriate access method, depending on particular application needs, is a rather challenging problem. A spatial index library that can combine a wide range of indexing techniques under a common application programming interface can thus prove to be a valuable tool, since it will enable efficient application integration of a variety of structures in a consistent and straightforward way.

The major difficulty with designing such a tool is that most index structures have a wide range of distinctive characteristics, that are difficult to compromise under a common framework. For example, some structures employ data partitioning while others use space partitioning, some have rectangular node types while others have spherical node types, some are balanced while other are not, some are used only for indexing points while others are better for rectangles, lines, or polygons. Another important issue is that the index structures have to provide functionality for exploiting the semantics of application-specific data types, through easy customization while making sure that meaningful queries can still be formulated for these types. Moreover, it is crucial to adopt a common programming interface in order to promote reusability, easier maintenance and code familiarity, especially for large application projects where many developers are involved. The framework should capture the most important design characteristics, common to most structures, into a concise set of interfaces. This will help developers concentrate on other aspects of the client applications promoting, in this manner, faster and easier implementation. The interfaces should be easily extensible in order to address future needs without necessitating revisions to client code. These fundamental requirements make the design of a *generic* spatial index framework a challenging task. Even though there is a substantial volume of work on spatial index structures and their properties, little work has appeared that addresses design and implementation issues.

The most relevant spatial index library to SaIL is the eXtensible and fleXible Library (XXL) [7]. XXL offers both low-level and high-level components for development and integration of spatial index structures like cursors, access to raw disk, a query optimizer, etc. Even though XXL is a superset of SaIL, it differs in three respects: First, SaIL offers a very concise, straightforward interface for querying arbitrary index structures in a uniform manner. In contrast, XXL querying interfaces are index specific (apart for join and aggregation queries). Second, SaIL offers more generic querying capabilities. Despite the fact that XXL can support a variety of advanced spatial queries (with the use of a framework for generalizing an incremental best-first search query strategy), nonconventional user defined queries have to be implemented by hand requiring modifications in all affected index structures. In contrast, SaIL offers an intuitive interface, utilizing well known design patterns, for formulating novel queries without having to revise the library in any way. Finally, SaIL provides the capability to customize query behavior during execution with the use of standardized design patterns. In contrast, in XXL this capability has to be supported explicitly by all index structure implementations.

GiST (for *Generalized Search Tree* [16]) is also relevant to this work. GiST is a framework that generalizes a height balanced, single rooted search tree with variable fanout. In essence, GiST is a *parameterized* tree that can be customized with user defined data types and user defined functions on these types which help guide the structural and searching behavior of the tree. Each node in the tree consists of a set of predicate/pointer pairs. Pointers are used to link the node to children nodes or data entries. Predicates are the user defined data types stored in the tree. The user, apart from choosing a predicate domain (e. g., the set of natural numbers, rectangles on a unit square universe, etc.), must also implement a number of methods (i. e., *consistent*, *union*, *penalty* and *pickSplit*) which are used internally by GiST to control the behavior of the tree. By using a simple interface, GiST can support a wide variety of search trees and their corresponding querying capabilities, including B-trees [5] and R-trees [13]. In order to support an even wider variety of structures, Aoki [1] proposed three additions to GiST. The GiST interface was augmented with multiple predicate support, which is useful for storing meta-data in the tree nodes. The tree traversal interface was also improved, so that the user can define complex search strategies. Finally, support for divergence control was added, so that a predicate contained in a parent node need not correspond to an accurate description of its subtree (for example an R-tree with relaxed parent MBRs that do not tightly enclose their children). Other extensions have also been proposed that modify the internal structure of GiST so that it can support very specialized indices, for instance, the TPR-tree [18] for moving objects. Aref and Ilyas proposed the SP-GiST framework [2] which provides a novel set of external interfaces and original design, for furnishing a generalized index structure that can be customized to support a large class of spatial indices with diverse structural and behavioral characteristics. The generality of SP-GiST allows the realization of space and data driven partitioning, balanced and unbalanced structures. SP-GiST can support k-D-trees [3], tries [6,10], quadtrees [9,19] and their variants, among others.

The design of SaIL is orthogonal to XXL, GiST, SP-GiST and their variants. These libraries address the implementation issues behind new access methods by removing for the developer the burden of writing structural maintenance code. SaIL does not aim to simplify the development process of the index structures per se, but more importantly, the development of the applications that use them. *In that respect, it can be used in combination with all other index structure developer frameworks*. Employing SaIL is an easy process that requires the implementation of simple adapter classes that will make index structures developed with XXL, GiST, and SP-GiST compliant to the interfaces of SaIL, conflating these two conflicting design viewpoints. Ostensibly, existing libraries can be used for simplifying client code development as well—and share, indeed, some similarities with SaIL (especially in the interfaces for inserting and deleting elements from the indices)—but given that they are not targeted primarily for that purpose, they are not easily extensible (without the subsequent need for client code revisions), they do not promote transparent usage of diverse indices (since they use index specific interfaces), and they cannot be easily used in tandem with any other index library (existing or not). Finally, SaIL is the first library that introduces functionality for supporting temporal and spatio-temporal indices, through specialized capabilities that all other libraries are lacking.

Scientific Fundamentals

The Architecture of SaIL

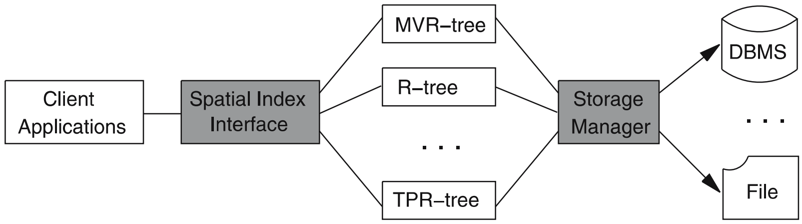
The SaIL framework consists of four components:

1. The core library toolkit which provides basic functionality. For example, generic types like variants, implementation of utility classes like property sets, an exception class hierarchy tailored for spatial indices, etc.
2. The storage manager toolkit for supporting diverse storage requirements. This container is useful for decoupling the index structure implementation from the actual storage system, enabling proper encapsulation.
3. The spatial index interface which is a set of abstractions of the most common spatial index operations and related types. For example, interfaces for nodes, leafs, spatial data types, spatio-temporal queries, and more.
4. The concrete implementations of various spatial index structures. For example, variants of the R-tree, the MVR-tree, the TPR-tree, etc.

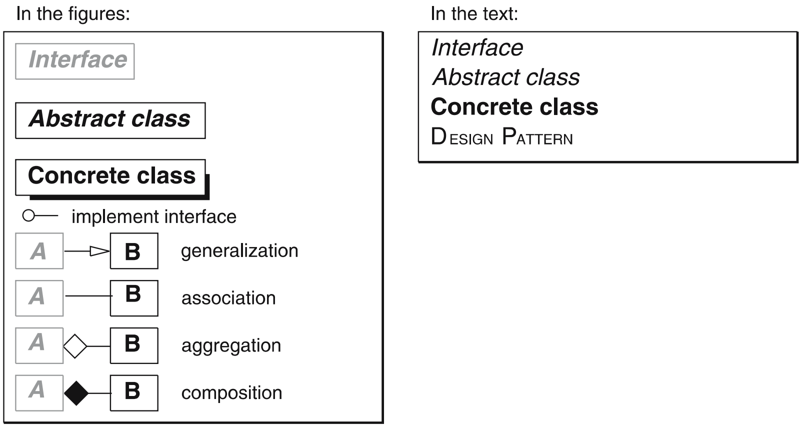
SaIL is designed to sit in-between the client application and the access methods that need to be interfaced by the client code. A simple architectural diagram is shown in Fig. 1. This section presents the most important concepts behind SaIL's design decisions using various examples. Figure 2 summarizes the UML notation used in the text and diagrams. When referring to a specific design pattern the definitions of Gamma et al. are used [12]. For convenience, some design pattern descriptions (quoted from [12]) are listed in Table 1.

The Core Toolkit The abstract and concrete classes offered by the core toolkit are shown in Fig. 3 (note that this and the remaining figures adopt the diagrammatic notation shown in Fig. 2). This toolkit addresses very simple but essential needs for any generic framework.

It provides a **Variant** type for representing a variety of different primitive types (like integers, floats, character arrays, etc.), which is necessary for avoiding hard coding



Index Structures, Extensible, Figure 1 Architectural diagram of SaIL



Index Structures, Extensible, Figure 2
Notation used in the paper

Index Structures, Extensible, Table 1 Design Pattern Descriptions

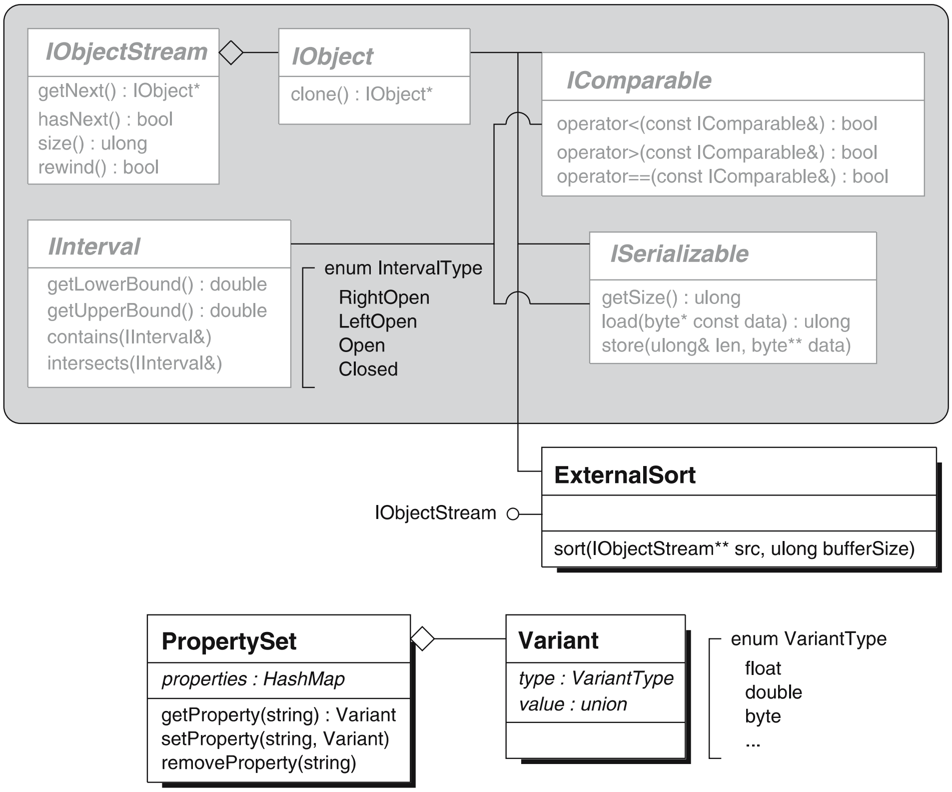
Name	Description
Memento	Without violating encapsulation, capture and externalize an object's internal state so that the object can be restored to this state later
Proxy	Provide a surrogate or place holder for another object to control access to it
Composite	Composite lets clients treat individual objects and compositions of objects uniformly
Facade	Provide a unified interface to a set of interfaces in a subsystem. Facade defines a higher-level interface that makes the subsystem easier to use
Visitor	Represent an operation to be performed on the elements of an object structure. Visitor lets you define a new operation without changing the classes of the elements on which it operates
Strategy	Define a family of algorithms, encapsulate each one, and make them interchangeable
Command	Encapsulate a request as an object, thereby letting you parameterize clients with different requests

specific primitive types in interface definitions that might need to be modified at a later time. It offers a **PropertySet**, or a collection of <*PropertyName, Value*> pairs. Property sets are useful for passing an indeterminate number of parameters to a method, even after the interfaces have been defined, without the need to extend them. For example, adding and handling the initialization of new properties to an object can be achieved without modifying its constructor, if a **PropertySet** is used as one of the constructor's arguments.

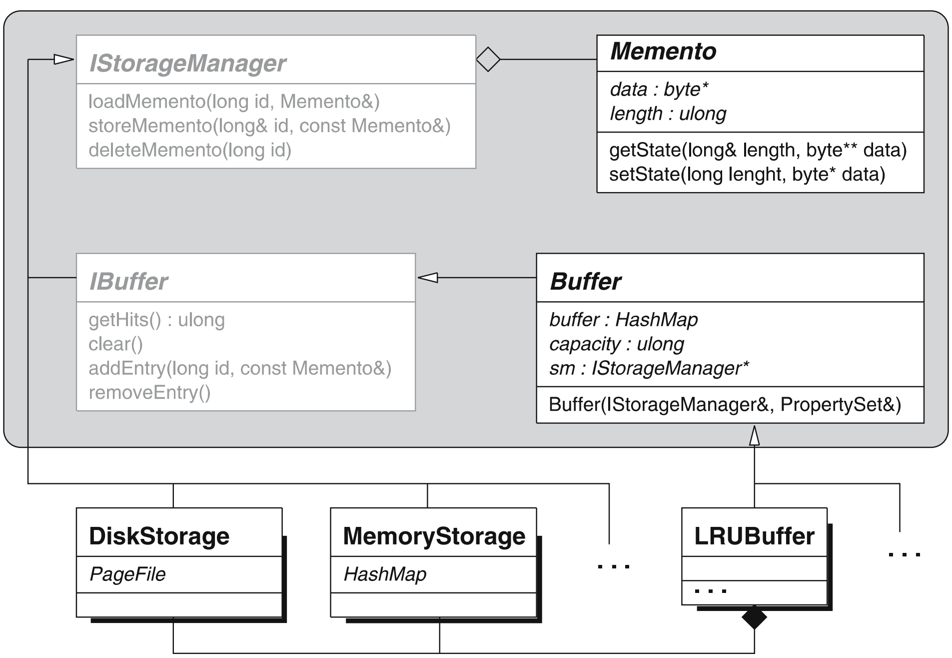
An index specific *Exception* class hierarchy is provided which helps promote the use of exception handling in client code, in a structured manner. Some basic interfaces representing essential object properties are also defined. For example interfaces for serializing and comparing objects, defining streams/iterators on object collections, etc. Other helper classes are provided as well, representing open/closed intervals, random number generators, resource usage utilities, etc. (some utilities are left out of this figure).

An important utility class provided by the core toolkit is **ExternalSort**. External sorting is needed for sorting very large relations that cannot be stored in main memory, an operation that is essential for bulk loading an index structure. By using the *IObject*, *IComparable* and *ISerializable* interfaces, **ExternalSort** is a generic sorting utility that can sort entries of any user defined data type in a very robust and straightforward manner. The caller needs to define an object stream that provides the sorter with objects of type *IObject* (using the *IObjectStream* interface), in random sequence. Then, **ExternalSort** iterates over the input objects is sorted order, as implied by a user provided comparator implementation. Sorting is accomplished by using temporary files on secondary storage. Conveniently, **ExternalSort** implements the *IObjectStream* interface itself and, thus, can be used as a Proxy in place of an *IObjectStream*, with the only difference that the stream appears to be in sorted order.

The Storage Manager Toolkit A critical part of spatial indexing tools is the storage manager, which should be versatile, very efficient and provide loose coupling. Clients that must persist entities to secondary storage should be unaware of the underlying mechanisms, in order to achieve proper encapsulation. Persistence could be over the network, on a disk drive, in a relational table, etc. All mediums should be treated uniformly in client code, in order to promote flexibility and facilitate the improvement of stor-



Index Structures, Extensible, Figure 3 The Core Toolkit



Index Structures, Extensible, Figure 4 The Storage Manager Toolkit

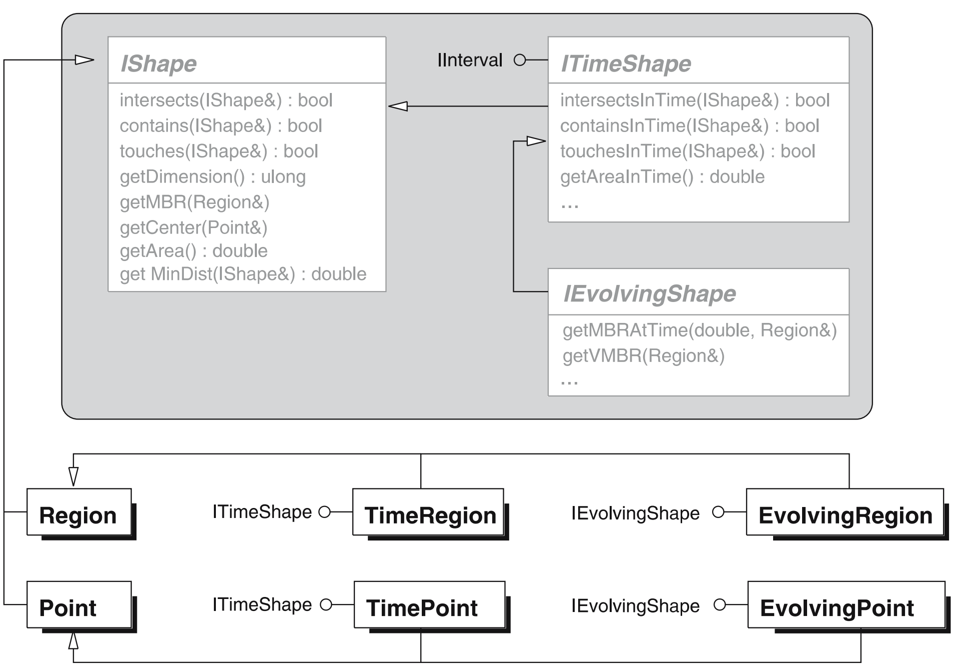
age management services as the system evolves, without the need to update the client code.

The storage manager toolkit is shown in Fig. 4. The key abstraction is a MEMENTO pattern that allows loose coupling between the objects that are persisted and the concrete implementation of the actual storage manager. An object that wants to store itself has to instantiate a concrete subclass of Memento that accurately represents its state. Then, it can pass this instance to a component supporting the *IStorageManager* interface, which will return an identifier that can be used to retrieve the object's state at a later time. Two concrete implementations of the *IStorageManager* interface are already provided. A main memory manager that uses a hash table and a disk based page file.

This architecture allows multiple layers of storage management facilities to be streamlined one after the other. A simple example is the need to store data over a network. A simple adapter class can be implemented that sends the data over the network with the proper format expected by the remote computer that uses a second adapter class to persist the data. Another example is the implementation of a relational based index structure. An adapter can be embedded on an existing implementation, converting the data to be persisted into appropriate SQL statements that store them in a relational table (e. g., as BLOBs).

The *IBuffer* interface provides basic buffering functionality. It declares the two most important operations of a buffer: adding and removing an entry. The *Buffer* abstract class provides a default implementation for the major operations of a buffer component. For convenience, a concrete implementation of a Least Recently Used buffering policy is already provided. Using the *IBuffer* interface is straightforward; it acts as a proxy between an actual storage manager and a client, buffering entries as it sees fit. The client instantiates a concrete buffer class, provides the buffer with a reference to the actual storage manager and finally associates the index structure (or any other component that will use the storage manager) with a reference to the buffer. Conveniently, the callers are unaware that buffering is taking place by assuming that a direct interface with the actual storage manager is used. This architecture provides sufficient flexibility to alter buffering policies at runtime, add new policies transparently, etc.

The Spatial Index Interface Spatial access methods are used for indexing complex spatial objects with varying shapes. In order to make the interfaces generic it is essential to have a basic shape abstraction that can also represent composite shapes and other decorations (meta-data like z-ordering, insertion time, etc.). The *IShape* COMPOSITE pattern (Fig. 5) is defined as an interface that all index structures should use to decouple their implementation from actual concrete shapes. For example, inserting convex polygons into an R-tree [13] can be accomplished by calling the *IShape.getMBR* method to obtain the minimum bounding region of the polygon. As long as the user defined polygon class returns a proper MBR representation, the R-tree remains unaware of the actual details of the polygon class, internally—all it needs to be aware of is the *IShape.getMBR* contract. Complex shapes and combinations of shapes can be represented by composing a number of *IShapes* under one class and, hence, can be handled in a uniform manner as all other classes of type *IShape*. Various methods useful for comparing *IShapes* (like *contains*, *intersects*, etc.) are also specified. In addition, the *IShape* interface can handle shapes with arbitrary dimensionality, so that multi-dimensional indices can be supported.



Index Structures, Extensible, Figure 5 The shape interfaces

The *ITimeShape* interface extends *IShape* with methods for comparing shapes with temporal predicates. An *ITimeShape* has temporal extents, acquired by implementing the *IInterval* interface. This abstraction is useful for handling spatio-temporal indices, where the indexed objects are associated with insertion and deletion times, as well as for specifying spatial queries with temporal restrictions.

Furthermore, to cover special cases where shapes evolve over time, the *IEvolvingShape* interface is provided. An evolving shape can be represented by associating a velocity vector to every vertex of the representation and, in addition, it can be approximated by an evolving MBR. Special methods must also be implemented for computing intersections, area, volume, and other characteristics over time. The *IEvolvingShape* interface is derived from *ITimeShape* and, thus, such shapes can be associated with temporal extents during which the evolution is taking place.

Concrete implementations of basic shapes are provided for convenience. These shapes provide basic functionality, like computing intersections, areas, temporal relations, etc.

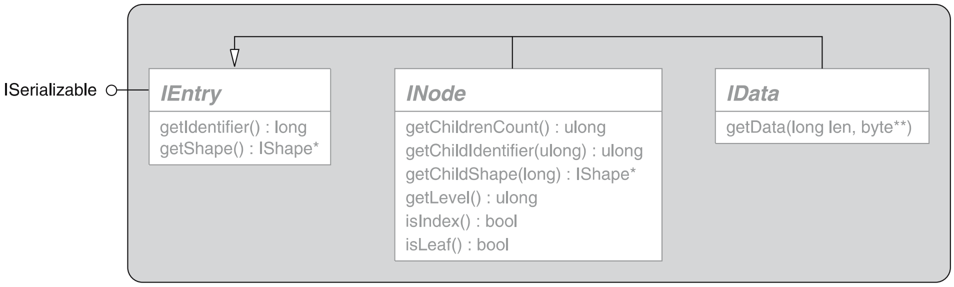
An essential capability of a generic index manipulation framework is to provide a sound set of index elements (leaf and index nodes, data elements, etc.) that enable consistent manipulation of diverse access methods from client code. For example, querying functions should return iterators (i. e., enumerations or cursors) over well-defined data elements, irrespective of what kind of structures they operate on. In general, a hierarchical index structure is composed of a number of nodes and a number of data entries. Every node has a specific shape, identifier and level. Nodes are further divided into index nodes and leaves. Data entries are either simple pointers to the actual data representations on disk (a secondary index), or the actual data themselves (a primary index). The data elements can be either exact shape representations or more generic meta-data associated with a specific index entry. These concepts are represented by using the following hierarchy: *IEntry* is the most basic interface for a spatial index entry; its basic members are an identifier and a shape. *INode* (that inherits from *IEntry*) represents a generic tree node; its basic members are the number of children, the tree level, and a property specifying if it is an index node or a leaf. The *IData* interface represents a data element and contains the meta-data associated with the entry or a pointer to the real data.

The core of the spatial index interface is the *ISpatialIndex* FACADE pattern. All index structures should implement *ISpatialIndex* (apart from their own custom methods), which abstracts the most common index operations. This interface is as generic as possible. All methods take *IShapes* as arguments and every shape is associated with a user defined identifier that enables convenient referencing of objects. Below each method is described in more detail.

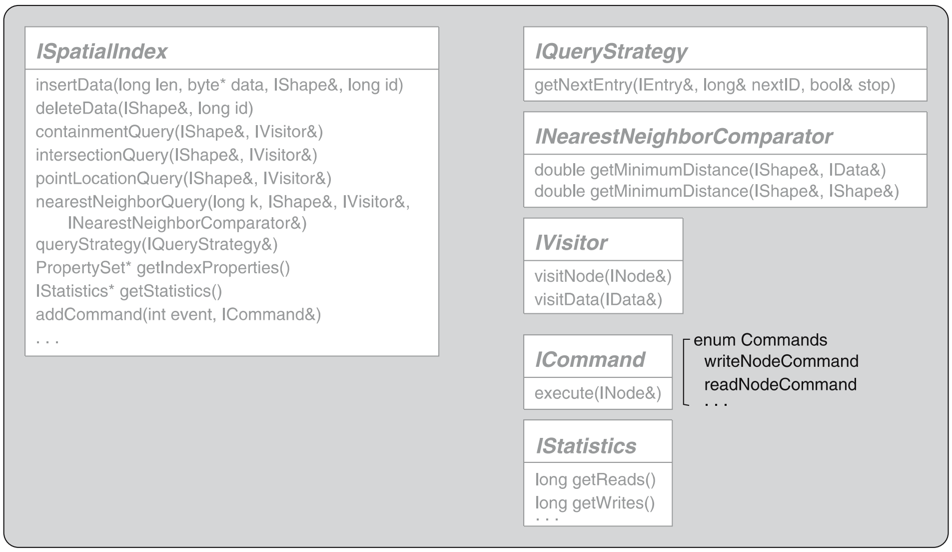
The *insertData* method is used for inserting new entries into an index. It accepts the data object to be inserted as an *IShape*—an interface that can be used as a simple decorator over the actual object implementation. Meta-data can also be stored along with the object as byte arrays (useful for implementing primary indices). The *deleteData* method locates and deletes an object already contained in an index. It accepts the *IShape* to be deleted and its object identifier. (The *IShape* argument is necessary since spatial indices cluster objects according to their spatial characteristics and not their identifiers.)

The query methods take the query *IShape* as an argument. This simple interface is powerful enough to allow the developer to create customized queries. For example, suppose a circular range query on an R-tree is required. Internally, the R-tree range search algorithm decides if a node should be examined by calling the query *intersects* predicate on a candidate node MBR. Hence, it suffices to define a **Circle** class that implements the *intersects* function (specific for intersections between circles and MBRs) and call the *intersectionQuery* method with a **Circle** object as its argument. Since arbitrarily complex shapes can be defined with the *IShape* interface, the querying capabilities of the index structures are only limited by the ability of the developer to implement correctly the appropriate predicate functions, for the *IShape* objects used as arguments to the querying methods.

In order to provide advanced customization capabilities a VISITOR pattern is used. The *IVisitor* interface is a very powerful feature. The query caller can implement an appropriate visitor that executes user defined operations when index entries are accessed. For example, the visitor can ignore all index and leaf nodes and cache all visited data entries (essentially the answers to the query) for later processing (like an enumeration). Instead, it could process the answers interactively (like a cursor), terminating the search when desired (an interactive k-nearest neighbor query is a good example, where the user can stop the execution of the query after an adequate number of answers has been discovered). Another useful example of the VISITOR pattern is tallying the number of query I/Os. By counting the number of times each visit method has been called, it is possible to count the number of index nodes, leaf nodes, or "any" level nodes that were accessed for answering a query (the *visitNode* method returns an *INode* reference when called, which provides all necessary information on the specific node accessed, e. g., its level, shape, etc.). A tallying visitor is presented in Algorithm 1. A different example is visualizing the progress of the query. As the querying algorithm proceeds, the visitor can draw the



Index Structures, Extensible, Figure 6 The index element interface



Index Structures, Extensible, Figure 7 The generic spatial index interface

last accessed node or data element on the screen. An access method can support the *IVisitor* interface simply by guaranteeing that all query algorithms call the *visitNode* and *visitData* methods of *IVisitor*, every time a node or a data entry is accessed while searching the structure. Thus, supporting the *IVisitor* interface requires a very simple, inexpensive procedure.

The *IShape* and *IVisitor* interfaces enable consistent and straightforward query integration into client code, increasing readability and extensibility. New index structures can add specialized functionality by requesting decorated *IShape* objects (thus, without affecting the interfaces). The *IVisitor* interface allows existing visitor implementations to be reused for querying different types of access methods and users can customize visitors during runtime.

To illustrate the simplicity of supporting the *IVisitor* interface from the querying methods of a spatial index implementation, the actual implementation of a range query algorithm of a hierarchical structure that supports all of the aforementioned features is shown in Algorithm 2.

An even larger degree of customization is provided for the nearest neighbor query method. Since different applications use diverse distance measures to identify nearest neighbors (like the Euclidean distance, and others),

```
class MyVisitor : public IVisitor {
public:
  map<long, IShape*> answers;
  long nodeAccesses;

  MyVisitor() : nodeAccesses(0) {}

  public visitNode(INode* n) {
    nodeAccesses++;
  }

  public visitData(IData* d) {
    // add the answer to the list.
    answers[d.getIdentifier()] = d.getShape();
  }
}
```

Index Structures, Extensible, Algorithm 1 *IVisitor* example

the *nearestNeighborQuery* method accepts an *INearestNeighborComparator* object. By allowing the caller to provide a customized comparator, the default nearest neighbor algorithm implemented by the underlying structure can be used, obviating any application specific changes to the library. In reality, a nearest neighbor comparator is

```
void rangeQuery(const IShape& query, IVisitor& v) {
  stack<NodePtr> st;
  Node* root = readNode(m_rootID);

  if (root->m_children > 0 && query.intersects(root->m_nodeBoundary)) st.push(root);

  while (! st.empty()) {
    Node* n = st.top(); st.pop();

    if (n->isLeaf()) {
      v.visitNode(*n);

      for (unsigned long cChild = 0; cChild < n->m_children; cChild++) {
        if (query.intersects(n->m_childBoundary[cChild])) {
          v.visitData(n->m_childData[cChild]);
        }
      }
    } else {
      v.visitNode(*n);

      for (unsigned long cChild = 0; cChild < n->m_children; cChild++)
        if (query.intersects(n->m_childBoundary[cChild]))
          st.push(readNode(n->m_childIdentifier[cChild]));
    }
  }
}
```

Index Structures, Extensible, Algorithm 2 Range query method implementation that supports the *IVisitor* interface

essential, since in order to find the actual nearest neighbors of a query, the query has to be compared with each candidate's exact representation so that an exact distance can be computed. Since most spatial index structures store object approximations in place of the real objects (e. g., R-trees store MBRs), internally they make decisions based on approximate distance computations and, hence, cannot identify the exact nearest neighbors. One way to overcome this weakness, is to let the index load the actual objects from storage and compute the real distances only when appropriate. Albeit, this would break encapsulation since loading the actual objects implies that the index has knowledge about the object representations. Alternatively, the user can provide a nearest neighbor comparator that implements a method for comparing index approximations (e. g., MBRs) with the actual objects (e. g., polygons). Method *getMinimumDistance* is used for that purpose.

For implementing "exotic" queries, without the need to make internal modifications to the library, a STRATEGY pattern is proposed. Using the *queryStrategy* method the caller can fully guide the traversal order and the operations performed on a structure's basic elements allowing, in effect, the construction of custom querying algorithms on the fly. This technique uses an *IQueryStrategy* object for encapsulating the traversal algorithm. The index structure calls *IQueryStrategy.getNextEntry* by starting the traversal from a root and the *IQueryStrategy* object chooses which entry should be accessed and returned next. The traversal can be terminated when desired. As an example, assume that the user wants to visualize all the index levels of an R-tree. Either the R-tree implementation should provide a custom tree traversal method that returns all nodes one by one, or a query strategy can be defined for the same purpose (which can actually be reused as is, or maybe with slight modifications, for any other hierarchical structure). An example of a breadth-first node traversal algorithm is presented in Algorithm 3 (the example requires less than 15 lines of code). Many other possible uses of the query strategy pattern exist.

Another capability that should be provided by most index structures is allowing users to customize various index operations (usually by the use of call-back functions). The spatial index interface uses a COMMAND pattern for that purpose. It declares the *ICommand* interface—objects implementing *ICommand* encapsulate user parameterized requests that can be run on specific events, like customized alerts. All access methods should provide a number of queues, each one corresponding to different events that trigger each request. For example, assume that a new index structure is being implemented. The function that persists

```
class MyQueryStrategy : public IQueryStrategy {
  queue<long> ids;
public:
  void getNextEntry(IEntry& e, long& nextID, bool& stop) {
    // process the entry.
    ...

    // if it is an index entry and not a leaf
    // add its children to the queue.
    INode* n = dynamic_cast<INode*>(&e);
    if (n != 0 && ! n->isLeaf())
      for (long cChild = 0; cChild < n->getChildrenCount(); cChild++)
        ids.push(n->getChildIdentifier(cChild));

    stop = true;
    if (! ids.empty()) {
      // if queue not empty fetch the next entry.
      nextID = ids.front(); ids.pop();
      stop = false;
    }
  }
};
```

Index Structures, Extensible, Algorithm 3 Breadth-first traversal of index nodes

a node to storage can be augmented with an empty list of *ICommand* objects. Using the *addCommand* method the user can add arbitrary command objects to this list, that get executed whenever this function is called, by specifying an appropriate event number (an enumeration is provided for that purpose). Every time the function is called, it iterates through the *ICommand* objects in the list and calls their execute method. Another example is the need to track specific index entries and be able to raise alerts whenever they take part in splitting or merging operations, or they get relocated a new disk page. The COMMAND pattern promotes reusability, clarity, and ease of extensibility without the need of subclassing or modifying the spatial index implementations simply to customize a few internal operations as dictated by user needs.

Key Applications

GIS and other applications that are related with spatial, spatio-temporal and, generally, multi-dimensional data can benefit significantly by using the SaIL framework for incorporating spatial and spatio-temporal index structures into existing code. For example, mapping applications [8], astronomical applications [20], traffic analysis and surveillance applications. A sample implementation in C++ and Java can be downloaded freely from [14].

Future Directions

The extensions to the framework for providing new functionality and supporting novel data structures and applications, are limitless. Currently, the prototype implementation includes R-tree variants [13], the MVR-tree [17], and the TPR-tree [18]. Exploring what are the necessary modifications for adapting the library to work with a diverse number of index structures that are not based on R-trees is an interesting avenue for future work.

Cross References

- ▶ Index Structures, Extensible
- ▶ Indexing, High Dimensional
- ▶ Indexing, Hilbert R-tree, Spatial Indexing, Multimedia Indexing
- ▶ Indexing the Positions of Continuously Moving Objects
- ▶ Indexing Schemes for Multi-dimensional Moving Objects
- ▶ Indexing Spatio-temporal Archives
- ▶ Nearest Neighbor Queries in Network Databases
- ▶ Nearest Neighbor Query
- ▶ Nearest Neighbors Problem
- ▶ R*-tree
- ▶ R-Trees – A Dynamic Index Structure for Spatial Searching

Recommended Reading

1. Aoki., P.M.: Generalizing "search" in generalized search trees (extended abstract). In: ICDE, pp. 380–389 (1998)
2. Aref, W.G., Ilyas, I.F.: An extensible index for spatial databases. In: SSDBM, pp. 49–58 (2001)
3. Bentley, J.L.: Multidimensional binary search trees used for associative searching. Communications of the ACM **18**(9):509–517 (1975)

4. Böm, C., Berchtold, S., Keim, D.A.: Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. ACM Computing Surveys **33**(3):322–373 (2001)
5. Comer, D.: The ubiquitous B-tree. ACM Computing Surveys **11**(2):121–137 (1979)
6. de la Briandais, R.: File searching using variable length keys. In: Proceedings of the Western Joint Computer Conference, pp. 295–298 (1959)
7. Van den Bercken, V., Blohsfeld, B., Dittrich, J., Krämer, J., Schäfer, T., Schneider, M., Seeger, B.: XXL – a library approach to supporting efficient implementations of advanced database queries. In: VLDB, pp. 39–48 (2001)
8. ESRI: ArcGIS. http://www.esri.com/software/arcgis/index.html
9. Finkel, R.A., Bentley, J.L.: Quad Trees, a data structure for retrieval on composite keys. Acta Informatica **4**(1):1–9 (1974)
10. Fredkin, E.: Trie memory. Communications of the ACM **3**(9):490–499 (1960)
11. Gaede, V., Günther, O.: Multidimensional access methods. ACM Computing Surveys **30**(2):170–231 (1998)
12. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional Computing Series. Addison-Wesley, New York, NY (1995)
13. Guttman, A.: R-trees: A dynamic index structure for spatial searching. In: SIGMOD, pp. 47–57 (1984)
14. Hadjieleftheriou, M.: SaIL. http://spatialindexlib.sourceforge.net
15. Hadjieleftheriou, M., Hoel, E., Tsotras, V.J.: Sail: A spatial index library for efficient application integration. GeoInformatica **9**(4):367–389 (2005)
16. Hellerstein, J.M., Naughton, J.F., Pfeffer, A.: Generalized search trees for database systems. In: VLDB, pp. 562–573 (1995)
17. Kumar, A., Tsotras, V.J., Faloutsos, C.: Designing access methods for bitemporal databases. TKDE **10**(1):1–20 (1998)
18. Saltenis, S., Jensen, C.S., Leutenegger, S.T., Lopez, M.A.: Indexing the positions of continuously moving objects. SIGMOD Record **29**(2):331–342 (2000)
19. Samet, H.: The quadtree and related hierarchical data structures. ACM Computing Surveys **16**(2):187–260 (1984)
20. SDSS. SkyServer. http://skyserver.sdss.org/dr1/en/

Indexing and Mining Time Series Data

EAMONN KEOGH
Computer Science & Engineering Department,
University of California – Riverside, Riverside, CA, USA

Synonyms

Similarity search; Query-by-content; Distance measures; Temporal data; Spatio-temporal indexing; Temporal indexing

Definition

Time series data is ubiquitous; large volumes of time series data are routinely created in geological and meteorological domains. Although statisticians have worked with time series for more than a century, many of their techniques hold little utility for researchers working with massive time series databases (for reasons discussed below). There two major areas of research on time series databases, the efficient discovery of previously *known* patterns (indexing), and the discovery of previously *unknown* patterns (data mining). As a concrete example of the former a user may wish to "*Find examples of a sudden increase, followed by slow decrease in lake volume anywhere in North America*" [14]. Such a query could be expressed in natural language, however virtually all indexing systems assume the user will sketch a query shape. In contrast, data mining aims to discover *previously unknown* patterns. For example "*Find all approximately repeated weekly patterns of vehicular traffic volume*". Because the space of unknown patterns is much larger than the space of known patterns, it should be obvious that data mining is a more demanding task in terms of both computer time and human intervention/interpretation.

Below are the major tasks considered by the time series data mining community. Note that indexing is sometimes considered a special case of data mining, and many data mining algorithms use indexing as a subroutine.

- **Indexing** (Query by Content/Similarity Search): Given a query time series Q of length n, a user defined query time series C of length m ($m \ll n$), and some similarity/dissimilarity measure $D(Q_{[i:i+m]},C)$, find the most similar time series in database DB [2,5,9,14].
- **Clustering**: Find natural groupings of the time series in database DB under some similarity/dissimilarity measure $D(Q,C)$ [4,10,12,20].
- **Classification**: Given an unlabeled time series Q, assign it to one of two or more predefined classes [6,12].
- **Motif Discovery**: Given an unlabeled time series Q of length n, and user defined subsequence length of m ($m \ll n$), find the pair of subsequences, A and B, that minimize $D(A,B)$.
- **Prediction** (Forecasting): Given a time series Q containing n datapoints, predict the value at time $n+1$.
- **Association Detection:** Given two or more time series, find relationships between them. Such relationships may or may not be casual and may or may not exist for the entire duration of the time series [3].
- **Summarization:** Given a time series Q containing n datapoints where n is an extremely large number, create a (possibly graphic) approximation of Q which retains its essential features but fits on a single page, computer screen etc [8,17].
- **Anomaly Detection** (Interestingness/Novelty Detection)**:** Given a time series Q, assumed to be normal, and an unannotated time series R. Find all sections of R which contain anomalies or "surprising/interesting/unexpected" occurrences [7,11,16].

I

- **Segmentation**: Given a time series Q containing *n* datapoints, construct a model $\overline{Q}$, from *K* piecewise segments ($K \ll n$) such that $\overline{Q}$ closely approximates Q [12]. Segmentation can be use to find regions of similar behavior, or simply to reduce the dimensionality of the data (see Fig. 3).

Historical Background

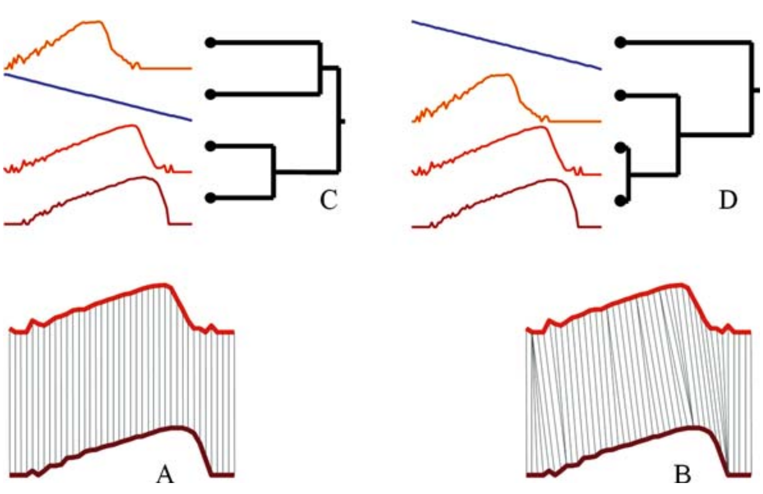
The task of indexing time series data can be traced back to a classic paper by Faloutsos, Ranganathan and Manolopoulos [5]. This paper also introduces the Gemini framework, which remains the basic framework for virtually all indexing (and many data mining) algorithms for time series. Given that most interesting datasets are too large to fit in main memory, the basic idea of the Gemini framework is to approximate the data in main memory, approximately solve the problem at hand, and then make (hopefully few) accesses to the disk to confirm or adjust the solution. Given this, a natural question to ask is which method should be used to approximate the data in main memory? The original paper suggested the discrete Fourier transform, but since then a bewilderingly large number of alternatives have been proposed. The section on Time Series Representations below considers this matter in some detail.

Scientific Fundamentals

Note that indexing, clustering and motif discovery make *explicit* use of a distance measure, and many approaches to classification, prediction, association detection, summarization and anomaly detection make *implicit* use of a distance measure. It is not surprising therefore, that the literature abounds with various distance measures, each with its proponents. Recently however, there has been an increasing understanding that the simple Euclidean distance is very difficult to beat in most domains [13]. One distance measure which *has* been shown to outperform Euclidean distance *some* datasets is Dynamic Time Warping (DTW). A visual intuition of both measures is shown in Fig. 1. Unlike Euclidean distance, which does a non-adaptive point-to-point alignment, DTW tries to find a natural peak-to-peak, valley-to-valley alignment of the data.

It is interesting to note that with the exception of indexing, research into the tasks enumerated above predate not only the decade old interest in data mining, but computing itself. What then, are the essential differences between the classic, and the data mining versions of these problems? The key difference is simply one of size and scalability; time series data miners routinely encounter datasets that are gigabytes in size. As a simple motivating example, consider hierarchical clustering. The technique has a long history, and well-documented utility. If however, in order to hierarchically cluster a mere million items, it would be necessary to construct a matrix with 10^{12} cells, well beyond the abilities of the average computer for many years to come. A data mining approach to clustering time series, in contrast, must explicitly consider the scalability of the algorithm [10].

In addition to the large volume of data, it is often the case that each individual time series has a very high dimensionality [2]. Whereas classic algorithms assume a relatively low dimensionality (for example, a few measurements such as "height, weight, blood sugar etc"), time series data mining algorithms must be able to deal with dimensionalities in the hundreds and thousands. The problems created by high dimensional data are more than mere computation time considerations, the very meanings of normally intuitive terms such as "similar to" and "cluster forming" become unclear in high dimensional space. The reason is that as dimensionality increases, all objects become essentially equidistant to each other, and thus classification



Indexing and Mining Time Series Data, Figure 1
A The Euclidean distance computes the similarity of two time series by comparing the i^{th} point of one with the i^{th} point another. **B** Dynamic Time Warping in contrast, allows non-linear alignments. For most domains, the DTW clustering produced by DTW (**D**) will be more intuitive than the clustering produced by Euclidean Distance (**C**)

and clustering lose their meaning. This surprising result is known as the "curse of dimensionality" and has been the subject of extensive research [1]. The key insight that allows meaningful time series data mining is that although the actual dimensionality may be high, the *intrinsic* dimensionality is typically much lower. For this reason, virtually all time series data mining algorithms avoid operating on the original "raw" data, instead they consider some higher-level representation or abstraction of the data.

Time Series Representations

As noted above, time series datasets are typically very large, for example, just a few hours of weather data can require in excess of a gigabyte of storage. This is a problem because for almost all data mining tasks, most of the execution time spent by algorithm is used simply to move data from disk into main memory. This is acknowledged as the major bottleneck in data mining, because many naïve algorithms require multiple accesses of the data. As a simple example, imagine attempting to do *k*-means clustering of a dataset that does not fit into main memory. In this case, every iteration of the algorithm will require that data in main memory to be swapped. This will result in an algorithm that is thousands of times slower than the main memory case.

With this in mind, a generic framework for time series data mining has emerged. The basic idea can be summarized as follows

Indexing and Mining Time Series Data, Table 1 A generic time series data mining approach

1)	Create an approximation of the data, which will fit in main memory, yet retains the essential features of interest
2)	Approximately solve the problem at hand in main memory
3)	Make (hopefully very few) accesses to the original data on disk to confirm the solution obtained in Step 2, or to modify the solution so it agrees with the solution obtained on the original datax

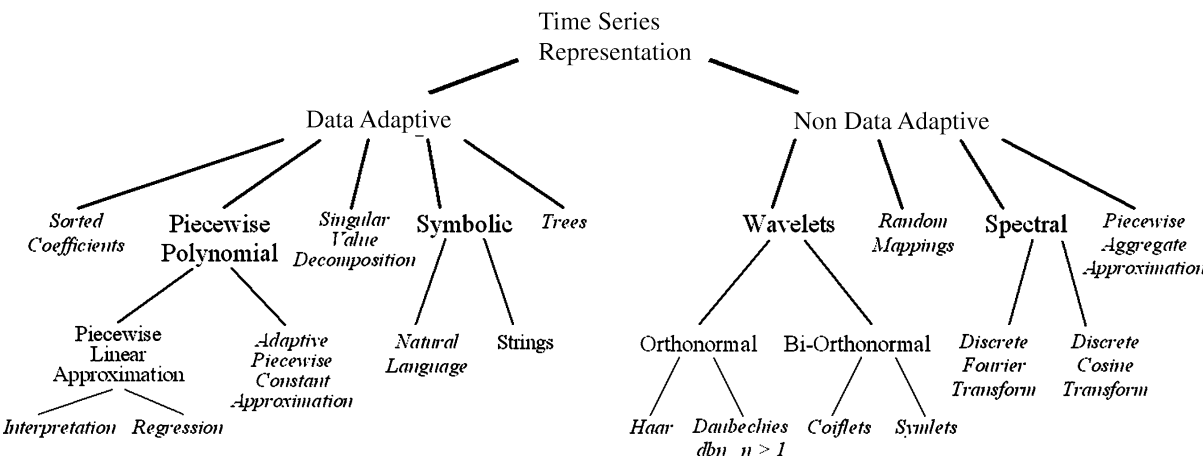
It should be clear that the utility of this framework depends heavily on the quality of the approximation created in Step 1. If the approximation is very faithful to the original data, then the solution obtained in main memory is likely to be the same, or very close to, the solution obtained on the original data. The handful of disk accesses made in Step 2 to confirm or slightly modify the solution will be inconsequential compared to the number of disks accesses required if the original data had been worked on. With this in mind, there has been a huge interest in approximate representation of time series. Figure 2 illustrates a hierarchy of every representation proposed in the literature.

To develop the reader's intuition about the various time series representations, Fig. 3 illustrates four of the most popular representations.

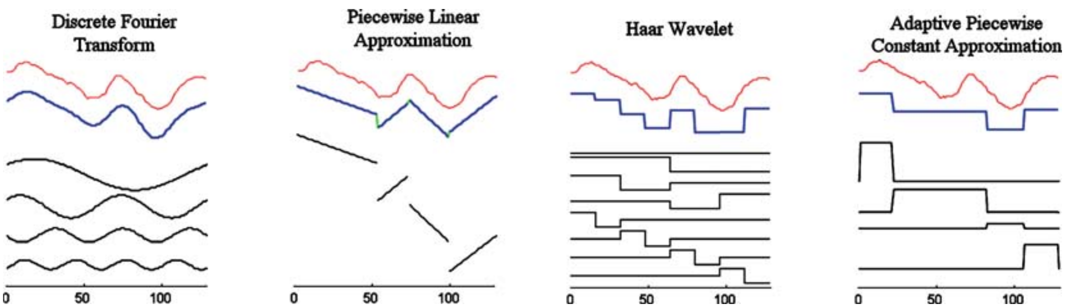
The Gemini framework discussed in the Historical Background section requires one special property of a time series representation in order to guarantee that it returns the true answer [5]. It must be the case that the distance function in the reduced dimensionality space underestimates the distance that would have be calculated in the original data space, the so called *Lower Bounding Lemma* [5,21]. This property has now been demonstrated for most representations, the exceptions being natural language, trees, random mappings and interpolation based Piecewise Linear Approximation.

Given the plethora of different representations, it is natural to ask which is best. Recall that the more faithful the approximation, the less clarification disks accesses will be needed to make in Step 3 of Table 1. In the example shown in Fig. 2, the discrete Fourier approach seems to model the original data the best, however it is easy to imagine other time series where another approach might work better. There have been many attempts to answer the question of which is the best representation, with proponents advocating their favorite technique [2,5,14,15]. The literature abounds with mutually contradictory statements such as "*Several wavelets outperform the … DFT*" [14], "*DFT-based and DWT-based techniques yield comparable results*" [18], "*Haar wavelets perform … better that DFT*" [9]. However an extensive empirical comparison on 50 diverse datasets suggests that while some datasets favor a particular approach, overall there is little difference between the various approaches in terms of their ability to approximate the data [13]. There are however, other important differences in the usability of each approach [2]. Below, some representative examples of strengths and weaknesses are considered.

The wavelet transform is often touted as an ideal representation for time series data mining, because the first few wavelet coefficients contain information about the overall shape of the sequence while the higher order coefficients contain information about localized trends [14,16]. This multiresolution property can be exploited by some algorithms, and contrasts with the Fourier representation in which every coefficient represents a contribution to the global trend [5,15]. However wavelets do have several drawbacks as a data mining representation. They are only defined for data whose length is an integer power of two. In contrast, the Piecewise Constant Approximation suggested by [19], has exactly the fidelity of resolution of as the Haar wavelet, but is defined for arbitrary length time series. In addition, it has several other useful properties such as the ability to support several different distance measures [19],



Indexing and Mining Time Series Data, Figure 2 A hierarchy of time series representations



Indexing and Mining Time Series Data, Figure 3 Four popular representations of time series. For each graphic is a raw time series of length 128. Below it is an approximation using 1/8 of the original space. In each case, the representation can be seen as a linear combination of basis functions. For example, the Discrete Fourier representation can be seen as a linear combination of the 4 sine/cosine waves shown in the bottom of the graphic

and the ability to be calculated in an incremental fashion as the data arrives [2]. Choosing the right representation for the task at hand is the key step in any time series datamining endeavor. The points above only serve as a sample of the issues that must be addressed.

Key Applications

This volume has numerous detailed articles on both indexing and mining time series (or spatial time series) data. The reader should consult (indexing), High-dimensional Indexing, Indexing Schemes for Multi-Dimensional Moving Objects, Extensible Spatial and SpatioTemporal Index Structures; (data mining), Algorithms for Mining Co-location Patterns, Co-location Pattern Discovery, Correlation Queries in Spatial Time Series Data, Discovering Similar Trajectories Using A Pseudo-Metric Distance Function.

Future Directions

Most work in data mining assumed that the data was static, and the user thus had the ability to do batch processing. As the field is maturing there is an increasing understanding that in most real world situations the data continuously arrives. There is therefore an increasing effort to extend current algorithms in motif detection / novelty detection / clustering etc to the streaming data case. In many cases this forces us to abandon the hope of producing an exact answer; instead it is necessary to be content with some probabilistic guarantees.

Readings

The field of time series data mining is relatively new, and ever changing. Because of the length of journal publication delays, the most interesting and useful work tends to appear in top-tier conference proceedings. Interested readers are urged to consult the latest proceedings of the major conferences in the field. These include the ACM Knowledge Discovery in Data and Data Mining, IEEE International Conference on Data Mining and the IEEE International Conference on Data Engineering.

Cross References

- ► Approximation
- ► Indexing, High Dimensional

► Indexing Spatio-temporal Archives
► Nearest Neighbors Problem
► Patterns in Spatio-temporal Data
► Trajectories, Discovering Similar

Recommended Reading

1. Aggarwal, C., Hinneburg, A., Keim, D.A.: On the surprising behavior of distance metrics in high dimensional space. In: Proceedings of the 8th International Conference on Database Theory, London, 6 January 2001, pp. 420–434
2. Chakrabarti, K., Keogh, E.J., Pazzani, M., Mehrotra, S.: Locally adaptive dimensionality reduction for indexing large time series databases. ACM Trans. Datab. Syst. **27**(2), 188–228 (2002)
3. Das, G., Lin, K., Mannila, H., Renganathan, G., Smyth, P.: Rule discovery from time series. In: Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, New York, 27–31 August 1998, pp. 16–22
4. Debregeas, A., Hebrail, G.: Interactive interpretation of kohonen maps applied to curves. In: Proceedings of the 4th International Conference of Knowledge Discovery and Data Mining, New York, 27–31 August 1998, pp. 179–183
5. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time-series databases. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, Minneapolis, 25–27 May 1994, pp. 419–429
6. Geurts, P.: Pattern extraction for time series classification. In: Proceedings of Principles of Data Mining and Knowledge Discovery, 5th European Conference, Freiburg, Germany, 3–5 September 2001, pp. 115–127
7. Guralnik, V., Srivastava, J.: Event detection from time series data. In: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, 15–18 August 1999, pp. 33–42
8. Indyk, P., Koudas, N., Muthukrishnan, S.: Identifying representative trends in massive time series data sets using sketches. In: Proceedings of the 26th International Conference on Very Large Data Bases, Cairo, Egypt, 10–14 September 2000, pp. 363–372
9. Kahveci, T., Singh, A.: Variable length queries for time series data. In: Proceedings of the 17th International Conference on Data Engineering, Heidelberg, Germany, 2–6 April 2001, pp. 273–282
10. Kalpakis, K., Gada, D., Puttagunta, V.: Distance measures for effective clustering of ARIMA time-series. In: Proceedings of the IEEE International Conference on Data Mining, San Jose, 29 November–2 December 2001, pp. 273–280
11. Keogh, E., Lonardi, S., Chiu, W.: Finding Surprising Patterns in a Time Series Database In Linear Time and Space. In: The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, 23–26 July 2002, pp. 550–556
12. Keogh, E., Pazzani, M.: An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In: Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, New York, 27–31 August 1998, pp. 239–241
13. Keogh, E., Kasetty, S.: On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. In: The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, 23–26 July 2002, pp. 102–111
14. Meyer, S.C.: Analysis of base flow trends in urban streams, northeastern Illinois, USA. Hydrogeol. J. **13**: 871–885 (2005)
15. Popivanov, I., Miller, R.J.: Similarity search over time series data using wavelets. In: Proceedings of the 18th International Conference on Data Engineering, San Jose, 26 February–1 March 2002, pp. 212–221
16. Rafiei, D., Mendelzon, A.O.: Efficient retrieval of similar time sequences using DFT. In: Proceedings of the 5th International Conference on Foundations of Data Organization and Algorithms, Kobe, Japan, 12–13 November 1998
17. Shahabi, C., Tian, X., Zhao, W.: TSA-tree: a wavelet based approach to improve the efficiency of multi-level surprise and trend queries. In: Proceedings of the 12th International Conference on Scientific and Statistical Database Management, Berlin, Germany, 26–28 July 2000, pp. 55–68
18. van Wijk, J.J., van Selow, E.: Cluster and calendar-based visualization of time series data. In: Proceedings 1999 IEEE Symposium on Information Visualization, 25–26 October 1999, pp 4–9. IEEE Computer Society
19. Wu, Y., Agrawal, D., El Abbadi, A.: A comparison of DFT and DWT based similarity search in time-series databases. In: Proceedings of the 9th ACM CIKM International Conference on Information and Knowledge Management. McLean, 6–11 November 2000, pp. 488–495
20. Xi, X., Keogh, E.J., Shelton, C.R., Li, W., Ratanamahatana, C.A.: Fast time series classification using numerosity reduction. In: Cohen, W.W., Moore, A. (eds.): Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, 25–29 June 2006, pp. 1033–1040. ACM 2006
21. Yi, B., Faloutsos, C.: Fast time sequence indexing for arbitrary lp norms. In: Proceedings of the 26th International Conference on Very Large Databases, Cairo, Egypt, 10–14 September 2000, pp. 385–394

Indexing API, Spatial/Spatio-temporal

► Index Structures, Extensible

Indexing, BDual Tree

Man Lung Yiu[1], Yufei Tao[2], Nikos Mamoulis[3]

[1] Department of Computer Science, Aalborg University, Aalborg, Denmark

[2] Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong, China

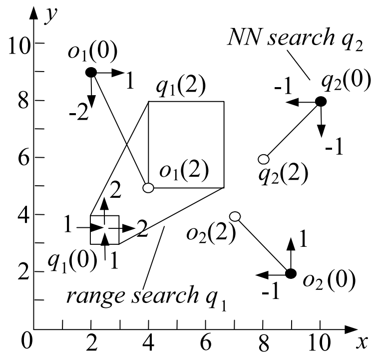
[3] Department of Computer Science, University of Hong Kong, Hong Kong, China

Synonyms

TPR-trees

Definition

The future location of a moving object is modeled as a linear function of time, consisting of its location at reference



Indexing, BDual Tree, Figure 1 Examples of spatiotemporal data and queries

time and its velocity. A centralized server manages the motion functions of all moving objects and an object issues an update to the server whenever its velocity changes. In Fig. 1, object o_1 is at location (2, 9) at time 0, and its velocities (represented with arrows) along the x- and y- dimensions are 1 and −2, respectively. A negative sign implies that the object is moving towards the negative direction of an axis.

A (predictive) *range query* returns the objects expected to appear (based on their motion parameters) in a moving rectangle q at some time within a future time interval qt. Figure 1 shows a query q_1 with $qt = [0, 2]$, whose extents at time 0 correspond to box $q_1(0)$. The left (right) edge of q_1 moves towards right at a velocity 1 (2), and the velocity of its upper (lower) boundary is 2 (1) on the y-dimension. Box $q_1(2)$ demonstrates the extents of q_1 at time 2. Notice that $q_1(2)$ has a larger size than $q_1(0)$ since the right (upper) edge of q_1 moves faster than the left (lower) one. The query result contains a single object o_1, whose location (4, 5) at time 2 falls in $q_1(2)$.

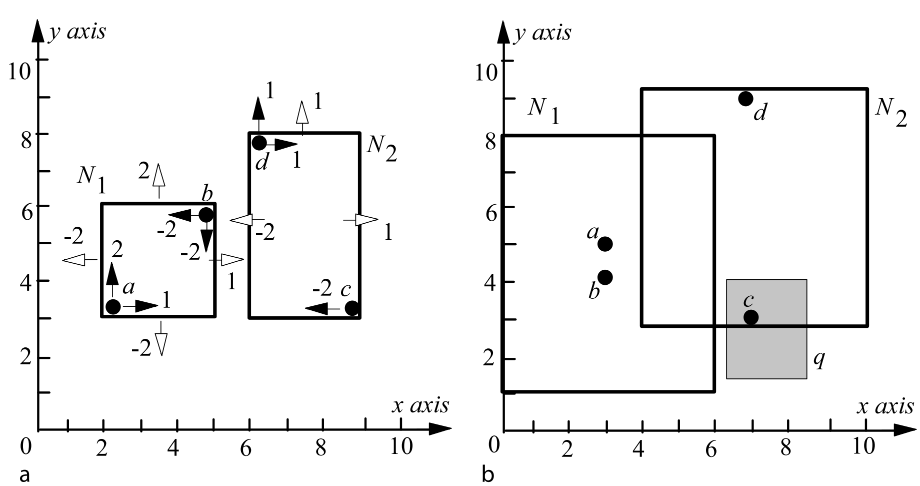
Various spatiotemporal indexes have been developed [3,4,5,6,7,9] to support efficiently (i) object updates, and (ii) query processing. However, they suffer from either large update cost or poor query performance. Motivated by this, another index called B^{dual}-tree [10] is designed to handle both updates and queries efficiently. B^{dual}-tree is a B^+-tree that indexes moving points according to their Hilbert values in the dual space defined by the location and velocity dimensions.

Historical Background

Existing predictive spatiotemporal structures can be classified into 3 categories: dual space indexes, time parameterized indexes, and space filling curve indexes.

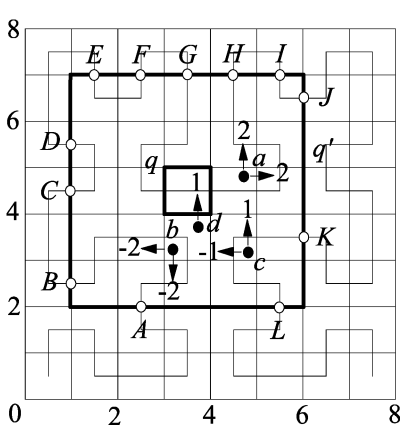
The *Hough-X representation* of a 2D moving point o is a vector $(o.v[1], o[1], o.v[2], o[2])$ where $o.v[i]$ is its velocity along dimension i $(1 \leq i \leq 2)$, and $o[i]$ is its i-th coordinate at the (past) reference time t_{ref}. Patel et al. [6] propose STRIPES, to index the 4D Hough-X representation of 2D moving points by a PR bucket quadtree. Object updates are processed fast because only a single path of the tree needs to be accessed. However, a leaf node in a quadtree may contain an arbitrarily small number of entries, and hence, more pages need to be accessed to obtain the same number of results. Although [6] suggest a "half-page" storage scheme to alleviate the problem, the query performance of STRIPES may still be high.

Saltenis et al. [7] propose the TPR-tree (later improved in [9]) that augments R-trees [2] with velocities to index moving objects. Figure 2a shows the locations of 4 objects at time 0, and the arrows indicate their movements. Figure 2b illustrates the object locations at time stamp 1. A node in the TPR-tree is represented as a *moving rectangle* (MOR), which includes (i) a SBox, a rectangle that tightly encloses the locations of the underlying objects at



Indexing, BDual Tree, Figure 2 A TPR-tree example. **a** SBox/VBox at time 0. **b** Node extents at time 1 time 0, and (ii) a VBox, a vector bounding their velocities. Observe that, in order to achieve good query performance, objects with similar motion parameters are grouped in the same TPR tree node. Consider a range query at time 1 whose search region q is the shaded rectangle in Fig. 2b. Since N_1 at time 1 does not intersect q, it does not contain any result, and can be pruned from further consideration. On the other hand, the query examines N_2, which contains the only qualifying object c. The MOR of a node grows with time such that it is guaranteed to enclose the locations of the underlying objects at any future time stamp, although it is not necessarily tight. Thus, during updates, MORs of nodes are tightened in order to optimize the performance of processing forthcoming queries. Due to the expensive node tightening triggered by object updates, TPR-tree is not feasible for real-life applications with frequent updates.

Jensen et al. [3] propose the B^x-tree, which consists of B^+-trees indexing the transformed 1D values of moving objects based on a space filling curve (e. g., Hilbert curve). Figure 3 shows an exemplary B^x-tree on 4 moving points. The location of an object at the reference time (e. g., 0) is mapped to a Hilbert value, which is indexed by a B^+-tree. Object updates are highly efficient by reusing the B^+ insertion/deletion procedures. Consider, for example, the small rectangle in Fig. 3 as a range query q at time stamp 1. First, q is expanded to another rectangle q' by the maximum object speed (i. e., 2) such that all actual results (i. e., any object intersecting q at time 1) are guaranteed to fall also in q' at time 0. Then, the region q' is decomposed into disjoint, consecutive Hilbert intervals and corresponding interval queries are executed on the B^+-tree for retrieving all points located inside q'. For each retrieved object, its actual location and velocity are verified against the original query q. Since expanding the query based on the maximum velocities of the entire dataset may lead to an excessively large number of false hits (e. g., objects a and b), the query performance of B^x-trees could be worser than that of TPR-trees.



Indexing, BDual Tree, Figure 3 A B^x-tree example

Scientific Fundamentals

With the exception of TPR-trees [7,9] (which optimize query performance by tightening nodes during updates), the query performance of the existing spatiotemporal indexes [3,6] degrades over time. In order to alleviate this problem, two trees (with different expiration time) are used for indexing objects alternatively. In the following, the above approach is adopted and two B^+-trees BT_1 and BT_2 are used alternatively for indexing objects. Interested readers may refer to [10] for details. Since BT_1 and BT_2 have the same structure, in the following, B^{dual}-tree is considered as a single tree.

A B^{dual}-tree has two parameters: (i) a *horizon* H, for deciding the farthest future time that can be efficiently queried, and (ii) a *reference time* T_{ref}, for converting moving points to their duals. A d-dimensional (in practice, $d = 2$ or 3) moving point o is represented with

- a *reference time stamp* $o.t_{ref}$ (e. g., the last update time of o),
- its coordinates $o[1], o[2], \ldots, o[d]$ at time $o.t_{ref}$, and
- its current velocities $o.v[1], o.v[2], \ldots, o.v[d]$.

For example, object o_1 in Fig. 1 has reference time $o_1.t_{ref} = 0$, coordinates $o_1[1] = 2$, $o_1[2] = 9$, and velocities $o_1.v[1] = 1$, $o_1.v[2] = -2$. The vector $o(t) = (o[1](t), o[2](t), \ldots, o[d](t))$ is used to denote the location of o at a time stamp $t \geq o.t_{ref}$, where, for $1 \leq i \leq d$:

$$o[i](t) = o[i] + o.v[i] \cdot (t - o.t_{ref}) \,. \tag{1}$$

The dual representation of an object o is a $2d$-dimensional vector:

$$o^{dual} = (o[1](T_{ref}), \ldots, o[d](T_{ref}), o.v[1], \ldots, o.v[d]) \,. \tag{2}$$

In other words, o^{dual} is a point in a $2d$-dimensional *dual space*, which contains d *location dimensions* $o[i](T_{ref})$ (for the first d components of o^{dual}) and d *velocity dimensions* $o.v[i]$. The dual space can be mapped to a 1D domain by any space filling curve. The Hilbert curve is chosen because it preserves the spatial locality better than other curves [3], achieving lower query cost.

Given the *resolution* λ (an integer), the dual space can be viewed as a *partitioning grid* of $2^{\lambda \cdot 2d}$ regular cells and each dimension has 2^{λ} divisions. Figure 4 depicts the dual space of 1D moving objects (i. e., $d = 1$) at resolution level $\lambda = 3$. The number in each cell represents its Hilbert value, which can be computed by a standard algorithm [1].

Objects whose duals o^{dual} fall in the same cell have identical Hilbert values, which are indexed by a B^+-tree, called the B^{dual}-tree. In the tree, each leaf entry stores the detailed dual representation of an object (i. e., locations at T_{ref}, and velocities).

Updates

Each object o issues an update whenever its velocity changes. Let $o^{\text{dual}}_{\text{old}}$ ($o^{\text{dual}}_{\text{new}}$) be the old (new) dual representation of the object. $o^{\text{dual}}_{\text{old}}$ is removed from the tree and then $o^{\text{dual}}_{\text{new}}$ is inserted into the tree. An insertion/deletion is performed in the same way as a B^+-tree, by accessing $O(\log N)$ pages where N is the dataset cardinality.

Specifying a Range Query

A d-dimensional moving rectangle (MOR) r is captured by

- a *reference time stamp* $r.t_{\text{ref}}$,
- a *spatial box* (SBox), a $2d$-dimensional vector ($r_{\vdash}[1]$, $r_{\dashv}[1], \ldots, r_{\vdash}[d]$, $r_{\dashv}[d]$), where [$r_{\vdash}[i]$, $r_{\dashv}[i]$] is the i-th ($1 \le i \le d$) projection of r at time $r.t_{\text{ref}}$, and
- a *velocity box* (VBox), a $2d$-dimensional vector ($r.V_{\vdash}[1]$, $r.V_{\dashv}[1], \ldots, r.V_{\vdash}[d]$, $r.V_{\dashv}[d]$), where $r.V_{\vdash}[i]$ (or $r.V_{\dashv}[i]$) indicates the velocity of the left (or right) edge on the i-th dimension.

Denoting the spatial extents of r at a time stamp $t \ge r.t_{\text{ref}}$ as $r(t) = (r_{\vdash}[1](t), r_{\dashv}[1](t), \ldots, r_{\vdash}[d](t), r_{\dashv}[d](t))$, is expressed by:

$$r_{\vdash}[i](t) = r_{\vdash}[i] + r.V_{\vdash}[i] \cdot (t - r.t_{\text{ref}})$$
$$r_{\dashv}[i](t) = r_{\dashv}[i] + r.V_{\dashv}[i] \cdot (t - r.t_{\text{ref}}) \, .$$

A range query specifies a time interval $qt = [qt_{\vdash}, qt_{\dashv}]$, and an MOR q whose reference time is $qt_{\vdash}$. For instance, for the range search in Fig. 1, $qt = [0, 2]$, and the query q_1 is an MOR with reference time 0, SBox (2, 3, 3, 4), and VBox (1, 2, 1, 2). An object o satisfies q if $o(t)$ falls in $q(t)$ for some $t \in qt$.

The Dual Space

Next, the dual space will be studied in more detail. Observe that any cell c in the partitioning grid can be regarded as a d-dimensional MOR (moving rectangle) whose SBox (VBox) captures the projection of the cell on the location (velocity) dimensions of the dual space. Figure 4 shows an example where $d = 1$, and the dual space has $2d = 2$ dimensions. The partitioning grid contains $2^{3 \cdot 2} = 64$ cells (i. e., the resolution $\lambda = 3$), and the number in each cell is the Hilbert value (of any point inside). The cell 53, for example, has a 1D SBox [0.5, 0.625] (its projection on the horizontal axis) and a VBox [0.375, 0.5], assuming that all the dimensions have a domain [0, 1].

1

21	22	25	26	37	38	41	42
20	23	24	27	36	39	40	43
19	18	29	28	35	34	45	44
16	17	30	31	32	33	46	47
15	12	11	10	53	52	51	48
14	13	8	9	54	55	50	49
1	2	7	6	57	56	61	62
0	3	4	5	58	59	60	63

0 1

Indexing, BDual Tree, Figure 4 Hilbert range decomposition ($d = 1$, $\lambda = 3$)

Given a range query q (an MOR), objects in a cell c need to be inspected if and only if the MOR of c intersects q at some time within the query interval qt. For example, assume $T_{\text{ref}} = 0$ and let c be the cell in Fig. 4 with value 53. According to the SBox and VBox of c, the spatial extent of c at time 1 is $c(1) = [0.5 + 0.375, 0.625 + 0.5] = [0.875, 1.125]$. For a query with $q = [0.7, 0.8]$, $q.V = [0.1, 0.1]$, and $qt = [0, 1]$, all the objects with Hilbert value 53 must be examined because $q(1) = [0.8, 0.9]$ intersects $c(1)$; otherwise, some actual results may be missed. The intersection algorithm in [7] can be applied to determine whether two MORs intersect at some time within a specified time interval.

Hilbert Interval Decomposition

In fact, as a property of the B^+-tree, an intermediate entry e is associated with an interval $[e.h_{\vdash}, e.h_{\dashv})$, which contains the Hilbert values of all the objects in the subtree. $[e.h_{\vdash}, e.h_{\dashv})$ is referred as the *Hilbert interval* of e. Each integer in the interval corresponds to a cell in the partitioning grid. As mentioned before, each cell can be regarded as an MOR, and thus, e can be trivially decomposed into $e.h_{\dashv} - e.h_{\vdash}$ MORs. However, the number of these MORs can be $2^{\lambda \cdot 2d}$ in the worst case (i. e., all the cells in the grid), such that the resulting query algorithms incur expensive CPU cost.

The goal is to break $[e.h_{\vdash}, e.h_{\dashv})$ into several disjoint intervals, such that the union of the cells in each interval is a hyper-square in the dual space. Figure 5 presents a hierarchical algorithm for decomposing a Hilbert interval (for a non-leaf entry e) into MORs. [10] proves the correctness of the algorithm and stated that an interval is decomposed into at most $(4^d - 1) \cdot (2\lambda - 1)$ MORs. Since $d = 2$ or 3 in most real applications, the computational cost is essentially linear to the resolution λ. In practice, with the typical parameter values $d = 2$ and $\lambda = 10$, the decomposition technique generates at most $(4^d - 1) \cdot (2\lambda - 1) = 285$

```
Algorithm Decompose(a Hilbert Interval HI)
S:=∅; // S will contain the decomposed perfect MORs eventually
r0:= the MOR covering the entire dual space;
ω0:= the interval of the Hilbert domain;
L:={(r0, ω0)}; // L is a FIFO queue
while (L is not empty)
   remove the first element (r, ω) of L; //r is a perfect MOR
   if (ω intersects HI) then
      if (ω is covered by HI) then
         add r to S;
      else if (the length of ω > 1) then
         divide r into 4^d identical perfect MORs;
         for each resulting MOR r' and its Hilbert interval ω'
            add (r', ω') to L;
return S;
```

Indexing, BDual Tree, Figure 5 Decomposing a Hilbert interval

MORs, which is much smaller than the number of MORs ($2^{\lambda 2d} = 1.1 \cdot 10^{12}$) produced by the trivial decomposition approach.

Figure 4 illustrates how the hierarchical decomposition algorithm works for the Hilbert interval [23, 49]. First, the 4×4 MORs of the dual space are considered: the intervals [0, 15], [16, 31], [32, 47], [48, 63]. A large MOR can be extracted from the interval [32, 47] because it is covered by [23, 49]. On the other hand, [23, 49] partially covers [16, 31]. Thus, the above technique is applied recursively on [16, 32] and consider its 2×2 MORs: the intervals [16, 19], [20, 23], [24, 27], [28, 31]. Now, two MORs can be extracted from the intervals [24, 27], [28, 31] because they are covered by [23, 49]. Again, [23, 49] partially covers [20, 23] so the 1×1 MORs of [20, 23] are considered and a MOR for [23,23] is extracted. Similarly, the other end of the interval [23, 49] can be decomposed into MORs for the intervals [48, 48], [49, 49].

In summary, the interval [23, 49] can be broken into 6 intervals [23, 23], [24, 27], [28, 31], [32, 47], [48, 48], [49, 49] satisfying the above condition. In particular, the cells in [23, 23], [24, 27], [32, 47] constitute 1×1, 2×2, and 4×4 squares, respectively. Each resulting square can be regarded as an MOR whose projection on a location/velocity dimension is identical to that of the square (e. g., [23, 49] can be associated with 6 MORs). Note that the actual number of MORs produced is usually much smaller than the upper bound $(4^d - 1) \cdot (2\lambda - 1)$ (e. g., the number 6 for the interval [23, 49] in Fig. 4 is much lower than the upper bound $(4^1 - 1) \cdot (2 \cdot 3 - 1) = 15$).

Query Processing

It is ready to discuss how to process a range query with a B^{dual}-tree. Let e be an intermediate entry, which can be decomposed into m MORs $r_1, r_2, \ldots, r_m$ ($m \leq (4^d - 1) \cdot (2\lambda - 1)$), by the algorithm of Fig. 5. Given a range query q, the subtree of e is pruned if no r_i ($1 \leq i \leq m$) intersects q during the query interval qt. The processing algorithm starts by checking, for each root entry e, whether any of its associated MORs intersects q during qt (in the same way as in TPR-trees [7]). If yes, the algorithm accesses the child node of e, carrying out the process recursively until a leaf node is reached. Then, detailed information of each object encountered is simply examined against the query predicate.

It is worth noticing that a B^{dual}-tree is as powerful as a TPR-tree in terms of the queries that can be supported. Intuitively, intermediate entries of both structures can be represented as MORs. Thus, an algorithm that applies to a TPR-tree can be adapted for the B^{dual}-tree. Adaptation is needed only because an entry of a TPR-tree has a single MOR, while that of a B^{dual}-tree corresponds to multiple ones. For instance, [10] discuss how B^{dual}-tree can be used to evaluate other more complex queries (e. g., predictive NN queries).

Key Applications

Range search is one of the most important operations in spatiotemporal databases. For instance, to perform flight control, an airport must continuously keep track of the aircrafts about to enter its vicinity in near future; this can be achieved by executing a range query periodically: "report the aircrafts that, in 10 minutes, will appear in the circle centering at the control tower with a radius of 20 miles". In a highway traffic monitoring system, on the other hand, a useful range query would "return the vehicles that are expected to enter Washington DC in 5 minutes". Besides being a useful stand-alone operator, range search is also the building block for complex retrieval tasks. For example, given a set of aircrafts, a *self distance join* would "find all pairs of aircrafts that will be within 50 miles from each other in 10 minutes". A strategy to process the join is to issue a range query for each aircraft. Specifically, the query region is a moving circle, which always centers at the aircraft and has a radius of 50 miles.

Future Directions

Like B^x-tree, the B^{dual}-tree assumes that each moving object is a point moving with constant velocity. However, the above assumption may not hold for all real-life applications.

Moving Objects with Extent

In practice, a moving object may refer to an object with extent (e. g., rectangle, circle). The B^{dual}-tree indexes moving points by mapping them to points in the dual space. Thus, it may not be directly applicable for moving objects with extents. It remains an open question whether an effective mapping technique can be developed such that a moving object (with extent) can be mapped to a single point in a transformed space.

Non-linear Moving Objects

Also, in the B^{dual}-tree, a point is assumed to be moving linearly with constant velocity. For instance, if an object (e. g., car) accelerates/decelerates or moves with circular motion (e. g., along the curve in a race track), then the object has to issue a large number of updates to the server. A better spatiotemporal index not only reduces the effort spent by the server, but also allows the objects to save energy by reducing their update frequencies. [8] generalizes TPR-trees by modeling the motion of an object as a motion matrix instead of a linear motion function. It is interesting to study how B^{dual}-tree can be extended for indexing non-linear moving objects such that the number of updates can be reduced.

Cross References

► Indexing of Moving Objects, B^x-Tree
► Space-Filling Curves

References

1. Butz, A.R.: Alternative Algorithm for Hilbert's Space-Filling Curve. IEEE Trans. Comput. **C-20**(4), 424–426 (1971)
2. Guttman, A.: R-trees: A Dynamic Index Structure for Spatial Searching. In: Proc. of ACM Management of Data (SIGMOD) (1984)
3. Jensen, C.S., Lin, D., Ooi, B.C.: Query and Update Efficient B^+-Tree Based Indexing of Moving Objects. In: Proc. of Very Large Data Bases (VLDB) (2004)
4. Kollios, G., Gunopulos, D., Tsotras, V.J.: On Indexing Mobile Objects. In: Proc. of ACM Symposium on Principles of Database Systems (PODS) (1999)
5. Kollios, G., Papadopoulos, D., Gunopulos, D., Tsotras, V.J.: Indexing Mobile Objects Using Dual Transformations. The VLDB Journal **14**(2):238–256 (2005)
6. Patel, J.M., Chen, Y., Chakka, V.P.: STRIPES: An Efficient Index for Predicted Trajectories. In: Proc. of ACM Management of Data (SIGMOD) (2004)
7. Saltenis, S., Jensen, C.S., Leutenegger, S.T., Lopez, M.A.: Indexing the Positions of Continuously Moving Objects. In: Proc. of ACM Management of Data (SIGMOD) (2000)
8. Tao, Y., Faloutsos, C., Papadias, D., Liu, B.: Prediction and Indexing of Moving Objects with Unknown Motion Patterns. In: Proc. of ACM Management of Data (SIGMOD) (2004)
9. Tao, Y., Papadias, D., Sun, J.: The TPR*-Tree: An Optimized Spatio-temporal Access Method for Predictive Queries. In: Proc. of Very Large Data Bases (VLDB) (2003)
10. Yiu, M.L., Tao, Y., Mamoulis, N.: The B^{dual}-Tree: Indexing Moving Objects by Space-Filling Curves in the Dual Space. The VLDB Journal, To Appear

Indexing Framework, Spatial/Spatio-temporal

► Index Structures, Extensible

Indexing, High Dimensional

MICHAEL GIBAS, HAKAN FERHATOSMANOGLU
Department of Computer Science and Engineering,
The Ohio State University, Columbus, OH, USA

Synonyms

Multi-dimensional indexing; Multi-dimensional access structures; Indexing, metric-space; Quantization; Time-series; VA-file; M-tree

Definition

High-dimensional indexing covers a number of techniques that are intended to offer faster access to high-dimensional data sets than a traditional sequential scan of the data itself. An index can provide more efficient query response by pruning the data search space. It is possible that increasing the dimensionality of an index can prune more search space at the cost of a larger index structure and more complicated index traversal. However, when the index dimensionality is too large, traditional multi-dimensional access structures are no longer effective. Techniques were developed that addressed the performance issues associated with high-dimensional indexes by 1) modifying existing techniques to expand the range of dimensionality for which they would be effective, 2) indexing objects based on characteristics of interest and 3) reading all data in much smaller quantized form.

Historical Background

As memory became less expensive, useful data sets became much larger in terms of both the raw size of the data and in terms of the number of attributes (dimensions) stored for each record. Additionally, data exploration tasks became more complicated. Complex objects were represented by *feature vectors*, a translation of the object into a series of attribute values. Objects were compared according to their similarity, a measure of closeness of one object's feature vector to another. Due to the size of the data sets it was too costly to read all records in the data set in order to answer a query. Due to the dimensionality of the data sets, the existing multi-dimensional access structures were not effective in efficiently answering queries. This led to the development of many techniques targeted toward addressing efficient query processing for large, high-dimensional data sets.

A data hierarchical multi-dimensional access structure, the R-tree, was first introduced in [8]. Many structures such as the X-tree [3] attempted to mitigate the performance issues associated with the R-tree at higher dimensionality. Other techniques were introduced [5] that index a metric abstraction of the data in order to efficiently answer a specific query type like nearest neighbor. Because of the severe performance degradation of multi-dimensional access structures at high-dimensionality, the VA-file was introduced [15] that essentially speeds up sequential scan.

Scientific Fundamentals

Typical Query Types

Index structure processing and effectiveness is partially dependent on query type. So first some common query types are discussed. A point query determines if a specific point exists in the database. A range query provides a set of points that exist within a given subspace. Similarity or Nearest Neighbor-type queries determine a set of points within the database that are 'closest' to the query point with respect to some distance function, such as Euclidean distance. Without indexing support, the point query can be answered by checking if any point matches the point query criteria. The range query is answered by determining which data objects have attribute values that fall within all the query criteria. The nearest neighbor query can be addressed by computing the distance function of each data object and keeping track of the closest ones.

Multi-dimensional Access Structures

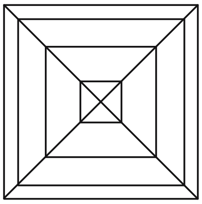
Multi-dimensional access structures were developed in order to take advantage of additional pruning that could take place in multiple dimensions that could not occur in a single dimension. A wide range of techniques are described in surveys about multi-dimensional indexing structures [4,7]. The structures typically fall into 2 categories, hierarchical data-partitioning structures and space partitioning structures. These structures can be used to answer the aforementioned common query types. Point queries can be answered by examining the data points falling within the partition(s) that contain the query point and checking for equality compared to the query point. Range queries are answered by examining the data points in all partitions that intersect with the space represented by the range and checking that they meet the query criteria. Nearest neighbor queries can be answered by keeping track of the best points examined thusfar and pruning those partitions that can not yield a better result.

These structures can work well when indexing a few dimensions, however they break down with respect to performance when the indexed dimensionality becomes too high. This is a phenomenon known as the curse of dimensionality. The data partitioning techniques break down because at higher dimensions, the hyper-shapes that could contain query matching objects increasingly overlap, and in order to find all potential query matches nearly the entire structure needs to be traversed. Non-overlapping space partitioning techniques break down because of increased size of the index structure at higher dimensions. As the dimensionality increases, the number of empty partitions increases exponentially.

Techniques have been introduced to mitigate the effect of the curse of dimensionality. Because data-hierarchical access structures degenerate at higher dimensions due to overlaps in the hyperspaces that are searched in order to find query matches, the X-tree [3] reduces these overlaps by allowing oversize nodes. This mitigates the effect of needing to search all the subspaces that overlap the query region at the cost of longer linear scan times of the supernodes. This structure can be effective for higher dimensions than other hierarchical data-partitioning structures, but still performs worse than sequential scan as the dimensionality becomes large enough.

The pyramid-technique [2] is a partitioning technique where the data space is subdivided into non-overlapping subspaces. Figure 1 shows a sample subdivision of data space in two dimensions. After dividing the space, the d dimensional space is transformed into a 1-dimensional space, which can be indexed using a B+-tree [1]. A range query is processed by examining data points in partitions that intersect with the query region. The technique does not suffer degraded performance as dimensionality increases when the query regions are not skewed. However, performance issues do arise when the query regions and data are skewed. There is also not a straightforward way to perform

I



Indexing, High Dimensional, Figure 1 Sample Pyramid-Technique Space Partitioning in 2-D

similarity type queries, since data points that are very close could be contained in partitions that are not close to each other.

Metric Space Indexing

Because many common queries for high-dimensional data are based on finding nearest neighbor or similar objects to some query object, metric space indexes were developed to provide such query support. The general idea is to index objects based on distances from parent substructures distances between objects and substructures and maintain these in a tree-like index. When a nearest neighbor query is performed, the tree is traversed by pruning paths that can not match the query.

An example of a metric space indexing structure is the M-tree. The M-Tree [5] is a balanced tree structure that associates data points to nodes based on the distance of the point from the node. A node consists of a spatial point, a covering radius, and a distance to the parent node. The M-tree is constructed by locating and inserting new points into the most suitable node. The most suitable node is one whose covering radius increases the least as a result of inserting the point. In the case that point lies in multiple potential nodes it is placed in the spatially closest node. Nearest neighbor queries are performed by traversing the structure using a branch-and-bound technique [12]. A queue of active subtrees that could contain matching points is maintained. Nodes are examined in order of how promising they are according to some heuristic and the distance of the current nearest neighbor is dynamically updated. When this distance is updated, it may be possible to prune candidate subtrees that could not contain a point that beats this distance.

These techniques can be effective for similarity searches using a given distance function. However, the index is only effective for the for the subset of attributes and distance function over which it was built. A new index will need to be constructed for each distance function or subset of attributes that are being queried.

Quantization Based Structures

Another class of techniques for high-dimensional data access is based on quantizing data. These techniques quantize the data into an approximate, but much smaller representation. Processing works by examining the approximated records and determining a set of records that could match the query.

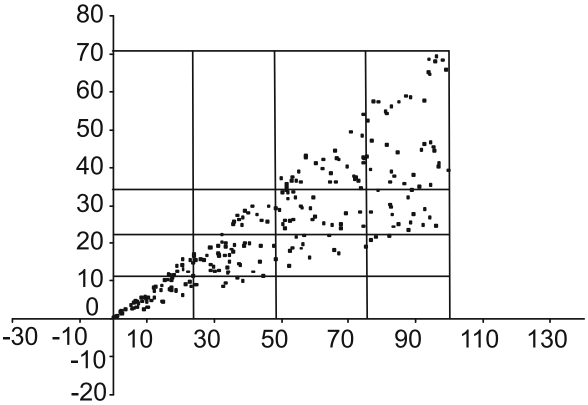
Indexing, High Dimensional, Table 1 Sample VA-File

Record ID	Attribute A	Attribute B	VA(A)	VA(B)
1	7.29	30.6	0000	0011
2	92.08	42.04	1011	0101
3	31.84	92.77	0011	1011
4	7.94	70.92	0000	1000
5	90.18	65.76	1011	1000
6	47.47	124.17	0101	1111
7	96.74	61.39	1100	0111
8	60.6	90.94	0111	1011
9	87.8	18.53	1010	0010
10	84.91	82.13	1010	1010
11	123.34	53.74	1111	0110
12	46.58	14.75	0101	0001
13	99.27	96.45	1100	1100
14	37.97	9.43	0100	0001
15	73.96	103.99	1001	1100
16	104.66	108.26	1101	1101
17	21.61	88.24	0010	1011
18	2.18	100.29	0000	1100
19	90.13	67.15	1011	1000
20	23.49	70.19	0010	1000

The VA-File (Vector Approximation) structure was the first such structure to follow this approach. Table 1 shows a simple example. Attributes A and B show the actual value for the object. Columns VA(A) and VA(B) show the vector approximation of the values. In this case the values are derived by using 4 bits for each attribute to divide the data space into equal width partitions.

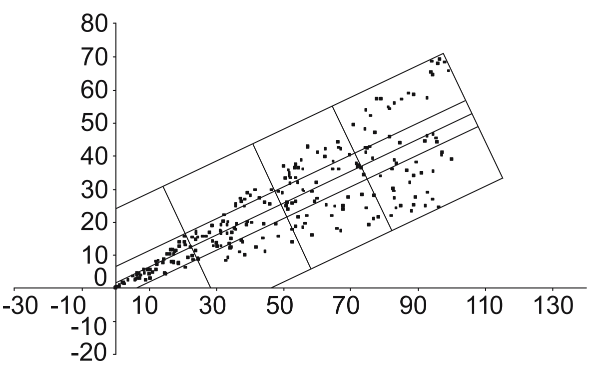
Point and range query processing works by first converting the query to cover the buckets that could answer the query. For example, if our query of interest was a range query where attribute A is between 37 and 47, and attribute B is between 10 and 20, the query would be converted to a VA query of VA(A) between 0100 and 0101 and VA(B) between 0001 and 0010. Then the VA-File would be traversed and any record's representation that intersected with the range queries approximated space would be identified. A second pass would go to disk to read the actual values and determine which of these records actually answered the query. In the sample query, objects 12 and 14 would meet the VA-file criteria with point 14 representing a false positive from the first pass that would be pruned in the second pass.

The effectiveness of the approximation associated with the VA-File is dependent on the resolution of data into distinct

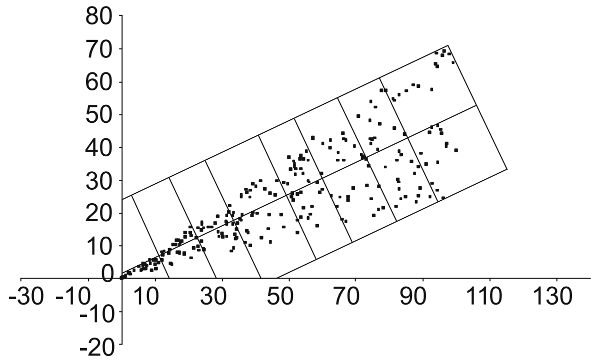


Indexing, High Dimensional, Figure 2 Sample VA-File Partitioning in 2-D

buckets. The VA-File does not take advantage of correlation between dimensions or the amount of energy associated with a given attribute. Figure 2 shows the cell representations for a sample VA-File. If two dimensions are highly correlated, there is little advantage with respect to pruning capability to index both dimensions. As well, if the attribute values for one dimension exhibit much more spread than the values of another dimension, it makes more sense to allocate more bits to differentiate the values in that dimension. The VA+-File [6] addresses both of these concerns by applying a quantizer that better differentiates data into meaningful cell representations. In the VA+-file, the data is first transformed using some technique to minimize the correlation between dimensions such as the Karhunen Loeve transform (KLT) [9,11]. Figure 3 displays a sample cell representation for 4 bit quantization for the same data shown for the VA-File. The budgeted space for a data object can also be better utilized by assigning bits to attributes non-uniformly. Figure 4 shows the same sample data set where 1 bit is assigned to the transformed y dimension and 3 bits are assigned to the transformed x-dimension. Another optimization that can be applied during VA+-file construction if appropriate for the data set is non-uniform quantization. Rather than using equi-populated, or equi-volumed splits, the cells can be split based on data clusters in each single dimension. Using Lloyd's algorithm [10] to determine the split points per dimension can help to keep data that is clustered in that dimension in the same bucket and improve the overall representation quality of the bucket.



Indexing, High Dimensional, Figure 3 Sample VA+-File Partitioning in 2-D, Uniform Bit Allocation



Indexing, High Dimensional, Figure 4 Sample VA+-File Partitioning in 2-D, Non-Uniform Bit Allocation

The advantage of these quantization-based structures is that performance does not get much worse at increased index dimensionality. However, a representation of each record must be visited, which can be time consuming if the number of records is very large or the quantized representation is not much smaller than the original data.

Key Applications

Characteristics of applications that can significantly benefit from high-dimensional indexing support include:

- data sets that are too large to read efficiently or to hold in memory
- data sets that cover many attributes
- applications that query objects based on many attributes (such as similarity queries or high-dimensional range queries)

Multimedia Databases

Multimedia databases are an application of interest with significant importance. Multimedia data is inherently large and is usually represented by a feature vector which describes the original data with a high number of dimensions. The similarity between two objects is defined by a distance function, e. g., Euclidean distance, between the corresponding feature vectors. A popular query, that makes use of the feature vectors, is the similarity query. For example, in image databases, the user may pose a query asking for the images most similar to a given image. Similarity query with multi-dimensional data is usually implemented by finding the k closest feature vectors to the feature vector of query object, which is known as k-nearest neighbor, k-NN, query. A closely related query is the ε-range query, where all feature vectors that are within ε neighborhood of the query point q are retrieved.

Geographic Databases

Geographic data sets can include the geographic spatial locations as well as attributes associated with objects. Additionally, the data objects themselves can be represented by complex geometric shapes in multiple-dimensions. Indexing techniques need to be able to associate an object with respect to its spatial characteristics as well as its other data characteristics.

Scientific Databases

Scientific databases are a typical example of applications that are both high-dimensional and for which high-dimensional queries are common. Evolution of computing technology has allowed detailed modeling of physical, chemical, and biological phenomena. Some examples from DOE scientific applications include climate simulation and model development, computational biology, high energy and nuclear physics, and astrophysics. Many large data sets in scientific domains contain a large number of attributes that may be queried and analyzed, and therefore considered as high dimensional data. For example, High Energy Physics data in one DOE scientific application contains more than 500 attributes that describe the properties of the objects in experiment data [14]. Various types of queries, such as partial match query and range query, are executed on these large data sets to retrieve useful information for scientific discovery. Astrophysics data has many of the same characteristics as geographic data. Data objects have spatial location or spatial shape as well as other attributes.

Biological Databases

Biological Databases are another example of high-dimensional data sets that can benefit from high-dimensional search support. An increasing number of biological databases, such as bio-sequence databases and biomedical data warehouses, are available online and have been used by many researchers to generate new knowledge. Queries asking sequence similarity are widely used to capture interesting and useful information from these bio-sequence databases. For example, the similarity of subsequences of a genome data to a query sequence is used to predict some diseases in advance, or to find some functional or physical relation between different organisms.

Text Databases

Many applications for document databases involve finding similar documents with respect to their intended meaning. As opposed to finding exact matches for certain search terms (as is common for many internet search engines), users could benefit from finding a set of documents reflecting the same context. Much like the image example, a text document can be converted to a high-dimensional representation of its contents. Similarity, or semantic closeness, can then be approximated by comparing the feature vectors of documents [13].

Time Series Data

The time series domain is another fertile area that can benefit from high-dimensional indexing. Many applications compare the similarity of objects with respect to attributes that reflect measurements taken at time intervals. Effective indexing of the time-series data allows efficient data exploration to find correlations and cause and effect relationships between data objects. Financial/business analysis and environmental monitoring are two such application domains.

Future Directions

Future solutions will endeavor to enhance performance for each of the classes of current techniques, both for data sets in general and also targeting specific query types for specific applications.

Cross References

► Indexing, X-Tree
► Nearest Neighbor Query
► Pyramid Technique
► R-Trees – A Dynamic Index Structure for Spatial Searching

Recommended Reading

1. Bayer, R., McCreight, E.: Organization and maintenance of large ordered indexes. In: Software Pioneers - Contributions to software engineering, pp. 245–262 (2002)
2. Berchtold, S., Böhm, C., Kriegel, H.-P.: The pyramid-tree: Breaking the curse of dimensionality. In: Haas, L.M., Tiwary, A. (eds.) SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, June 2–4, 1998, Seattle, Washington, USA, pp. 142–153. ACM Press, New York (1998)

3. Berchtold, S., Keim, D., Kriegel, H.-P.: The x-tree: An index structure for high-dimensional data. In: VLDB '96: Proceedings of the 22th International Conference on Very Large Data Bases, pp. 28–39. Morgan Kaufmann Publishers Inc, San Francisco, CA (1996)
4. Böhm, C., Berchtold, S., Keim, D.: Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. ACM Comput. Surv. **33**(3):322–373 (2001)
5. Ciaccia, P., Patella, M., Zezula, P.: M-tree: An efficient access method for similarity search in metric spaces. In: VLDB '97: Proceedings of the 23rd International Conference on Very Large Data Bases, pp. 426–435. Morgan Kaufmann Publishers Inc, San Francisco, CA (1997)
6. Ferhatosmanoglu, H., Tuncel, E., Agrawal, D., El Abbadi, A.: Vector approximation based indexing for non-uniform high dimensional data sets. In: CIKM '00: Proceedings of the ninth international conference on Information and knowledge management, pp. 202–209. ACM Press, New York, NY (2000)
7. Gaede, V., Günther, O.: Multidimensional access methods. ACM Comput. Surv. **30**(2):170–231 (1998)
8. Guttman, A.: R-trees: a dynamic index structure for spatial searching. In: SIGMOD '84: Proceedings of the 1984 ACM SIGMOD international conference on Management of data, pp. 47–57. ACM Press, New York, NY (1984)
9. Karhunen, H.: Über lineare Methoden in der Wahrscheinlichkeitsrechnung. Ann. Acad. Science Fenn (1947)
10. Lloyd, S.P.: Least squares quantization in pcm. IEEE Transactions on Information Theory, **28**(2), 127–135 (1982)
11. Loeve, M.: Fonctions aleatoires de seconde ordre. Processus Stochastiques et Mouvement Brownien, Gauthier-Villars, Paris (1948)
12. Roussopoulos, N., Kelley, S., Vincent, F.: Nearest neighbor queries. In: SIGMOD (1995)
13. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM **18**, 613–620 (1975)
14. SciDAC: Scientific data management center. http://sdm.lbl.gov/sdmcenter/ (2002)
15. Weber, R., Schek, H.-J., Blott, S.: A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In: VLDB (1998)

Indexing, Hilbert R-Tree, Spatial Indexing, Multimedia Indexing

IBRAHIM KAMEL
University of Sharjah, Sharjah, UAE

Synonyms

Database indexing; Multidimensional index; Spatial indexing; Multimedia indexing

Definition

Hilbert R-tree, an R-tree variant, is an index for multidimensional objects like lines, regions, 3-D objects, or high dimensional feature-based parametric objects. It can be thought of as an extension to B+-tree for multidimensional objects.

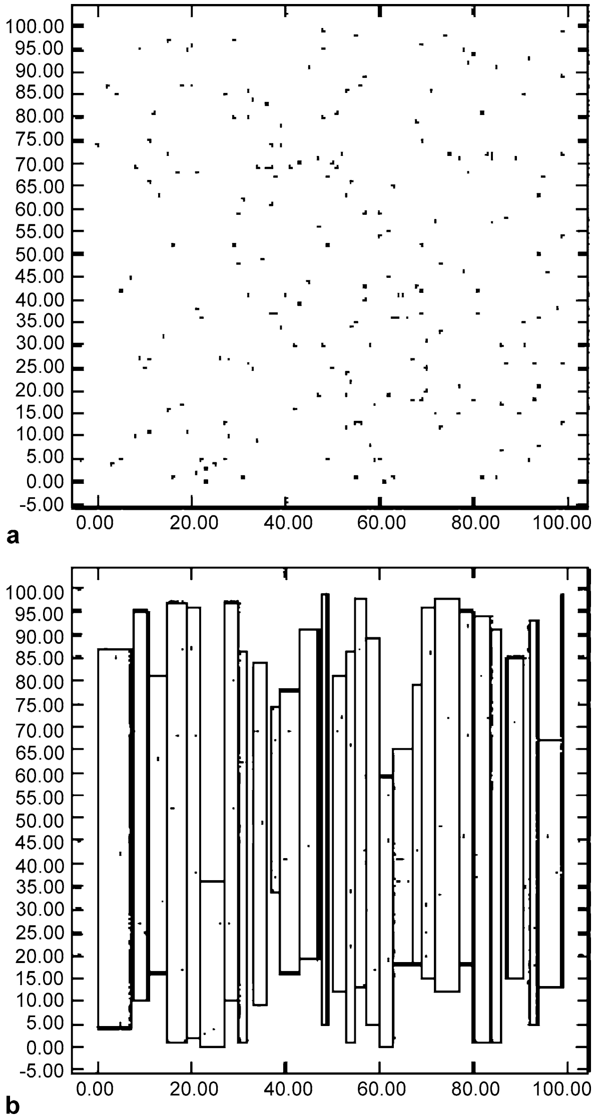
The performance of R-trees depends on the quality of the algorithm that clusters the data rectangles on a node. Hilbert R-trees use space filling curves, specifically the Hilbert curve, to impose a linear ordering on the data rectangles.

There are two types of Hilbert R-tree, one for static database and one for dynamic databases. In both cases, space filling curves and specifically the Hilbert curve are used to achieve better ordering of multidimensional objects in the node. This ordering has to be 'good' in the sense that it should group 'similar' data rectangles together to minimize the area and perimeter of the resulting minimum bounding rectangles (MBRs). Packed Hilbert R-trees are suitable for static databases in which updates are very rare or in which there are no updates at all.

The dynamic Hilbert R-tree is suitable for dynamic databases where insertions, deletions, or updates may occur in real time. Moreover, dynamic Hilbert R-trees employ a flexible deferred splitting mechanism to increase the space utilization. Every node has a well-defined set of sibling nodes. By adjusting the split policy, the Hilbert R-tree can achieve a degree of space utilization as high as is desired. This is done by proposing an ordering on the R-tree nodes. Hilbert R-tree sorts rectangles according to the Hilbert value of the center of the rectangles (i. e., MBR). Given the ordering, every node has a well-defined set of sibling nodes; thus, deferred splitting can be used. By adjusting the split policy, the Hilbert R-tree can achieve as high a utilization as desired. To the contrary, other R-tree variants have no control over the space utilization.

Historical Background

Although the following example is for a static environment, it explains the intuitive principals for good R-tree design. These principals are valid for both static and dynamic databases. Roussopoulos and Leifker proposed a method for building a packed R-tree that achieves almost 100% space utilization. The idea is to sort the data on the x or y coordinate of one of the corners of the rectangles. Sorting on any of the four coordinates gives similar results. In this discussion, points or rectangles are sorted on the x coordinate of the lower left corner of the rectangle. In the discussion below, the Roussopoulos and Leifker's method is referred to as the *lowx* packed R-tree. The sorted list of rectangles is scanned; successive rectangles are assigned to the same R-tree leaf node until that node is full; a new leaf node is then created and the scanning of the sorted list continues. Thus, the nodes of the resulting R-tree will be fully packed, with the possible exception of the last node at each level. Thus, the utilization is ≈100%. Higher levels of the tree are created in a similar way.



Indexing, Hilbert R-Tree, Spatial Indexing, Multimedia Indexing, Figure 1 **a** 200 points uniformly distributed. **b** MBR of nodes generated by the *lowx packed R-tree* algorithm

Figure 1 highlights the problem of the *lowx* packed R-tree. Figure 1**b** shows the leaf nodes of the R-tree that the *lowx* packing method will create for the points of Fig. 1**a**. The fact that the resulting father nodes cover little area explains why the *lowx* packed R-tree achieves excellent performance for point queries. However, the fact that the fathers have large perimeters, explains the degradation of performance for region queries. This is consistent with the analytical formulas for R-tree performance [7]. Intuitively, the packing algorithm should ideally assign nearby points to the same leaf node. Ignorance of the y- coordinate by the *lowx* packed R-tree tends to violate this empirical rule.

Scientific Fundamentals

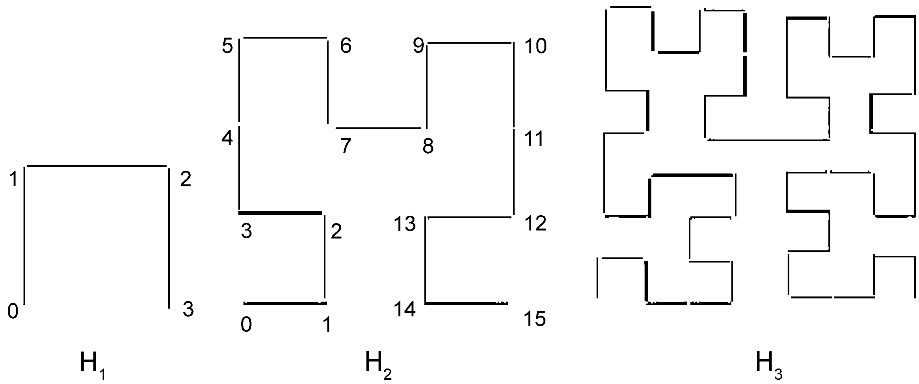
This section describes two variants of the Hilbert R-trees. The first index is suitable for the static database in which updates are very rare or in which there are no updates at all. The nodes of the resulting R-tree will be fully packed with the possible exception of the last node at each level. Thus, the space utilization is ≈100%; this structure is called a packed Hilbert R-tree. The second index supports insertions and deletions and is suitable for a dynamic environment, hence called the *Dynamic Hilbert R-tree*.

Packed Hilbert R-Trees

The following provides a brief introduction to the Hilbert curve. The basic Hilbert curve on a 2×2 grid, denoted by H_1, is shown in Fig. 2. To derive a curve of order i, each vertex of the basic curve is replaced by the curve of order $i-1$, which may be appropriately rotated and/or reflected. Figure 2 also shows the Hilbert curves of order two and three. When the order of the curve tends to infinity, like other space filling curves, the resulting curve is a *fractal* with a fractal dimension of two [5,7]. The Hilbert curve can be generalized for higher dimensionalities. Algorithms for drawing the two-dimensional curve of a given order can be found in [3,5]. An algorithm for higher dimensionalities is given in [2].

The path of a space filling curve imposes a linear ordering on the grid points; this path may be calculated by starting at one end of the curve and following the path to the other end. The actual coordinate values of each point can be calculated. However, for the Hilbert curve, this is much harder than, for example, the Z-order curve. Figure 2 shows one such ordering for a 4×4 grid (see curve H_2). For example, the point (0,0) on the H_2 curve has a Hilbert value of 0, while the point (1,1) has a Hilbert value of 2.

The Hilbert curve imposes a linear ordering on the data rectangles and then traverses the sorted list, assigning each set of C rectangles to a node in the R-tree. The final result is that the set of data rectangles on the same node will be close to each other in the linear ordering and most likely in the native space; thus, the resulting R-tree nodes will have smaller areas. Figure 2 illustrates the intuitive reasons why our Hilbert-based methods will result in good performance. The data is composed of points (the same points as given in Fig. 1). By grouping the points according to their Hilbert values, the MBRs of the resulting R-tree nodes tend to be small square-like rectangles. This indicates that the nodes will likely have a small area and small perimeters. Small area values result in good performance for point queries; small area and small perimeter values lead to good performance for larger queries.



Indexing, Hilbert R-Tree, Spatial Indexing, Multimedia Indexing, Figure 2 Hilbert curves of order 1, 2, and 3

Algorithm Hilbert-Pack:
(packs rectangles into an R-tree)
Step 1. Calculate the Hilbert value for each data rectangle
Step 2. Sort data rectangles on ascending Hilbert values
Step 3. */* Create leaf nodes (level l-0) */*
While (there are more rectangles)
generate a new R-tree node
assign the next C rectangles to this node
Step 4. */* Create nodes at higher level (l + 1) */*
While (there are > 1 nodes at level l)
sort nodes at level $l \geq 0$ on ascending
creation time
repeat Step 3

The assumption here is that the data are static or the frequency of modification is low. This is a simple heuristic for constructing an R-tree with 100% space utilization which at the same time will have as good of a response time as possible.

Dynamic Hilbert R-Trees

The performance of R-trees depends on the quality of the algorithm that clusters the data rectangles on a node. Hilbert R-trees use space-filling curves, specifically the Hilbert curve, to impose a linear ordering on the data rectangles. The Hilbert value of a rectangle is defined as the Hilbert value of its center.

Tree Structure

The Hilbert R-tree has the following structure. A leaf node contains at most C_l entries, each of the form *(R, obj_id)*, where C_l is the capacity of the leaf, R is the MBR of the real object $(x_{\text{low}}, x_{\text{high}}, y_{\text{low}}, y_{\text{high}})$ and *obj-id* is a pointer to the object description record. The main difference between the Hilbert R-tree and the R*-tree [1] is that non-leaf nodes also contain information about the LHVs. Thus, a non-leaf node in the Hilbert R-tree contains at most C_n entries of the form

$$(R, ptr, LHV)\,,$$

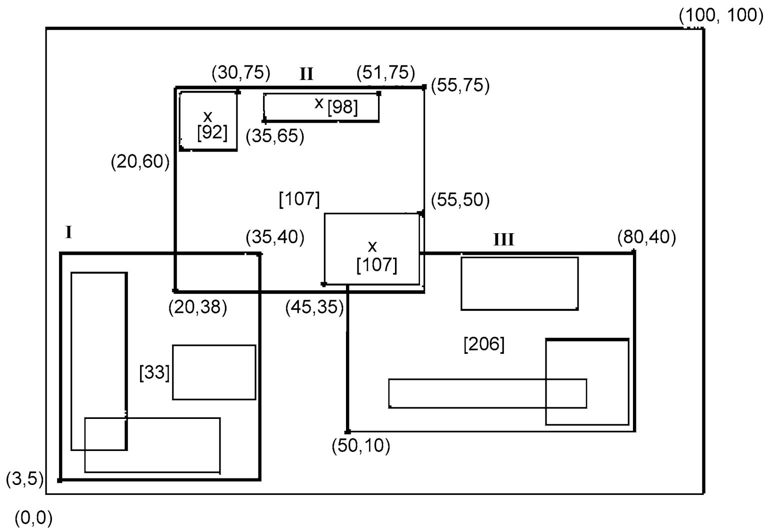
where C_n is the capacity of a non-leaf node, R is the MBR that encloses all the children of that node, ptr is a pointer to the child node, and LHV is the largest Hilbert value among the data rectangles enclosed by R. Notice that since the non-leaf node picks one of the Hilbert values of the children to be the value of its own LHV, there is no extra cost for calculating the Hilbert values of the MBR of non-leaf nodes. Figure 3 illustrates some rectangles organized in a Hilbert R-tree. The Hilbert values of the centers are the numbers near the 'x' symbols (shown only for the parent node 'II'). The LHV's are in [brackets]. Figure 4 shows how the tree of Fig. 3 is stored on the disk; the contents of the parent node 'II' are shown in more detail. Every data rectangle in node 'I' has a Hilbert value $v \leq 33$; similarly every rectangle in node 'II' has a Hilbert value greater than 33 and ≥ 107, etc.

A plain R-tree splits a node on overflow, creating two nodes from the original one. This policy is called a *1-to-2* splitting policy. It is possible also to defer the split, waiting until two nodes split into three. Note that this is similar to the B*-tree split policy. This method is referred to as the 2-to-3 splitting policy. In general, this can be extended to a s-to-$(s+1)$ splitting policy; where s is the order of the splitting policy. To implement the order-s splitting policy, the overflowing node tries to push some of its entries to one of its $s-1$ siblings; if all of them are full, then an s-to-$(s+1)$ split is required. The $s-1$ siblings are called the *cooperating siblings*.

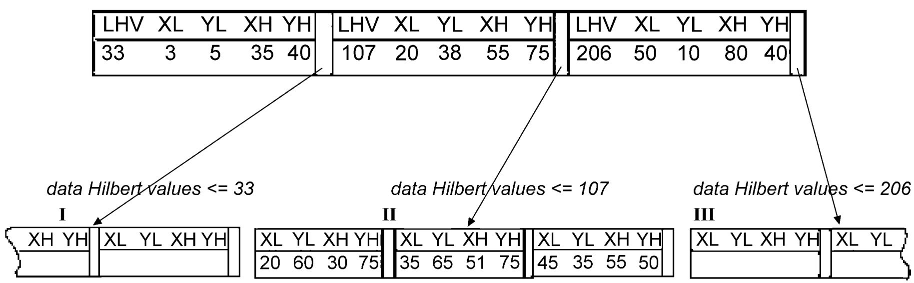
Next, the algorithms for searching, insertion, and overflow handling are described in detail.

Searching

The searching algorithm is similar to the one used in other R-tree variants. Starting from the root, it descends the tree and examines all nodes that intersect the query rectangle. At the leaf level, it reports all entries that intersect the query window w as qualified data items.



Indexing, Hilbert R-Tree, Spatial Indexing, Multimedia Indexing, Figure 3 Data rectangles organized in a Hilbert R-tree (Hilbert values and LHV's are in Brackets)



Indexing, Hilbert R-Tree, Spatial Indexing, Multimedia Indexing, Figure 4 The file structure for the Hilbert R-tree

Algorithm Search(node Root, rect w):
S1. *Search nonleaf nodes:*
Invoke Search for every entry whose MBR intersects the query window *w*.
S2. *Search leaf nodes:*
Report all entries that intersect the query window *w* as candidates.

Insertion

To insert a new rectangle *r* in the Hilbert R-tree, the Hilbert value h of the center of the new rectangle is used as a key. At each level, the node with the minimum LHV of all its siblings is chosen. When a leaf node is reached, the rectangle r is inserted in its correct order according to h. After a new rectangle is inserted in a leaf node N, **AdjustTree** is called to fix the MBR and LHV values in the upper-level nodes.

Algorithm Insert(node Root, rect r):
/* Inserts a new rectangle *r* in the Hilbert R-tree. *h* is the Hilbert value of the rectangle*/
I1. *Find the appropriate leaf node:*
Invoke **ChooseLeaf(r, h)** to select a leaf node L in which to place *r*.
I2. *Insert r in a leaf node L:*
If L has an empty slot, insert *r* in L in the appropriate place according to the Hilbert order and return.
If L is full, invoke **HandleOverflow(L,r)**, which will return new leaf if split was inevitable,
I3. *Propagate changes upward:*
Form a set S that contains L, its cooperating siblings and the new leaf (if any)
Invoke **AdjustTree(S)**.
I4. *Grow tree taller:*
If node split propagation caused the root to split, create a new root whose children are the two resulting nodes.

Algorithm ChooseLeaf(rect r, int h):
/* Returns the leaf node in which to place a new rectangle *r*. */
C1. *Initialize:*
Set *N* to be the root node.
C2. *Leaf check:*
If *N* is a leaf_return *N*.
C3. *Choose subtree:*
If *N* is a non-leaf node, choose the entry (R, ptr, LHV) with the minimum LHV value greater than *h*.
C4. *Descend until a leaf is reached:*
Set *N* to the node pointed by ptr and repeat from C2.

Algorithm AdjustTree(set S):
/* S is a set of nodes that contains the node being updated, its cooperating siblings (if overflow has occurred) and the newly created node NN (if split has occurred). The routine ascends from the leaf level towards the root, adjusting the MBR and LHV of nodes that cover the nodes in S. It propagates splits (if any) */
A1. If root level is reached, stop.
A2. *Propagate node split upward:*
Let N_p be the parent node of N.
If N has been split, let NN be the new node.
Insert NN in N_p in the correct order according to its Hilbert value if there is room. Otherwise, invoke HandleOverflow(N_p, NN).
If N_p is split, let PP be the new node.
A3. *Adjust the MBR's and LHV's in the parent level:*
Let P be the set of parent nodes for the nodes in S.
Adjust the corresponding MBR's and LHV's of the nodes in P appropriately.
A4. *Move up to next level:*
Let S become the set of parent nodes P, with NN = PP, if N_p was split.
repeat from A1.

Deletion

In the Hilbert R-tree there is no need to re-insert orphaned nodes whenever a father node underflows. Instead, keys can be borrowed from the siblings or the underflowing node is merged with its siblings. This is possible because the nodes have a clear ordering (according to Largest Hilbert Value, LHV); in contrast, in R-trees there is no such concept concerning sibling nodes. Notice that deletion operations require s cooperating siblings, while insertion operations require $s-1$ siblings.

Algorithm Delete(r):
D1. *Find the host leaf:*
Perform an exact match search to find the leaf node L that contains r.
D2. *Delete r :*
Remove r from node L.
D3. If L underflows
borrow some entries from s cooperating siblings.
if all the siblings are ready to underflow.
merge $s+1$ to s nodes,
adjust the resulting nodes.
D4. *Adjust MBR and LHV in parent levels.*
form a set S that contains L and its cooperating siblings (if underflow has occurred).
invoke **AdjustTree(S)**.

Overflow Handling

The overflow handling algorithm in the Hilbert R-tree treats the overflowing nodes either by moving some of the entries to one of the $s-1$ cooperating siblings or by splitting s nodes into $s+1$ nodes.

Algorithm HandleOverflow(node N, rect r):
/* return the new node if a split occurred. */
H1. Let ε be a set that contains all the entries from N and its $s-1$ cooperating siblings.
H2. Add r to ε.
H3. If at least one of the $s-1$ cooperating siblings is not full, distribute ε evenly among the s nodes according to Hilbert values.
H4. If all the s cooperating siblings are full, create a new node NN and distribute ε evenly among the $s+1$ nodes according to Hilbert values_return NN.

Key Applications

Hilbert R-tree is an index structure for multidimensional objects which commonly appear in Multimedia databases, Geographical Information Systems (GIS), and medical databases. For example, in multimedia databases, objects like images, voice, video, etc. need to be stored and retrieved. In GIS, maps contain multidimensional points, lines, and polygons, all of which are new data types.
Another example of such non-traditional data types can be found in medical databases which contain 3-dimensional brain scans (e. g., PET and MRI studies) For example, in these databases one of the common queries is "display the PET studies of 40-year old females that show high physiological activity inside the hippocampus." Temporal databases fit easily in the framework since time can be considered as one more dimension. Multidimensional objects appear even in traditional databases, for example, where a record with k attributes corresponds to a point in the k-space.

Future Directions

The performance of multi-dimensional indexes can be further improved through developing heuristics that produce smaller MBR for R-tree nodes. As it was shown in [7], the search performance of the R-trees improves by minimizing the perimeters and areas of the R-tree nodes. Grouping close-by multidimensional objects together in the same node would result in a better index.

Cross References

► Quadtree and Octree
► R*-tree

► R-Trees – A Dynamic Index Structure for Spatial Searching
► Space-Filling Curves

References

1. Beckmann, N., Kriegel, H., Schneider, R., Seeger, B.: The R*-tree: an efficient and robust access method for points and rectangles. In: Proc. of ACM SIGMOD, pp. 322–331. Atlantic City, NJ, May 1990
2. Bially, T.: Space-filling curves. Their generation and their application to bandwidth reduction. IEEE Trans. on Information Theory **15**(6), 658–664 (1969)
3. Griffiths, J.: An algorithm for displaying a class of space-filling curves. Software-Practice and Experience **16**(5), 403–411 (1986)
4. Guttman, A.: R-trees: a dynamic index structure for spatial searching. In: Proc. of ACM SIGMOD, pp. 47–57. Boston, MA, June 1984
5. Jagadish, H.: Linear clustering of objects with multiple attributes. In: Proc. of ACM SIGMOD Conf., pp. 332–342. Atlantic City, NJ, May 1990
6. Kamel, I., Faloutsos, C.: Parallel R-Trees. In: Proc. of ACM SIGMOD Conf., pp. 195–204. San Diego, CA, June 1992
7. Kamel, I., Faloutsos, C.: On Packing R-trees. In: Second International ACM Conference on Information and Knowledge Management (CIKM), pp. 490–499. Washington D.C., 1–5 Nov 1993
8. Kamel, I., Faloutsos, C.: Hilbert R-tree: An improved R-tree using fractals. In: Proc. of VLDB Conf., pp. 500–509. Santiago, Chile, September 1994
9. Koudas, N., Faloutsos, C., Kamel, I.: Declustering Spatial Databases on a Multi-Computer Architecture. In: International Conference on Extending Database Technology (EDBT), pp. 592–614. Avignon, France, 25–29 March 1996
10. Roussopoulos, N., Leifker, D.: Direct spatial search on pictorial databases using Packed R-trees. In: Proc. of ACM SIGMOD, pp. 17–31. Austin, TX, May 1985
11. Schroeder, M.: Fractals, Chaos, Power Laws: Minutes From an Infinite Paradise. W.H. Freeman and Company, NY (1991)
12. Sellis, T., Roussopoulos, N., Faloutsos, C.: The R+-Tree: a dynamic index for multi-dimensional objects. In: Proc. 13th International Conference on VLDB, pp. 507–518. England, September 1987

Indexing, Metric-Space

► Indexing, High Dimensional

Indexing, Mobile Object

► Mobile Object Indexing

Indexing Moving Objects

► Indexing the Positions of Continuously Moving Objects
► Mobile Object Indexing

Indexing Moving Points

► Indexing Schemes for Multi-dimensional Moving Objects

Indexing, Native Space

► Movement Patterns in Spatio-temporal Data

Indexing, Native-Space

► Indexing Spatio-temporal Archives

Indexing of Moving Objects, B^x-Tree

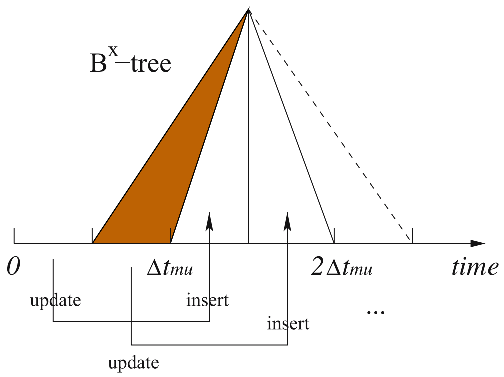
CHRISTIAN S. JENSEN[1], DAN LIN[2], BENG CHIN OOI[2]
[1] Department of Computer Science, Aalborg University, Aalborg, Denmark
[2] Department of Computer Science, National University of Singapore, Singapore, Singapore

Synonyms

B^x-tree; Moving objects; Linearization; Peano curve; Range query algorithm

Definition

The B^x-tree [1] is a query and update efficient B^+-tree-based index structure for moving objects which are represented as linear functions. The B^x-tree uses a linearization technique to exploit the volatility of the data values being indexed i. e., moving-object locations. Specifically, data values are first partitioned according to their update time and then linearized within the partitions according to a space-filling curve, e. g., the Peano or Hilbert curve. The resulting values are combined with their time partition information and then indexed by a single B^+-tree. Figure 1 shows an example of the B^x-tree with the number of index partitions equal to two within one maximum update interval Δt_{mu}. In this example, there are maximum of three partitions existing at the same time. After linearization, object locations inserted at time 0 are indexed in partition 1, object locations updated during time 0 to 0.5 Δt_{mu} are indexed in partition 2 and objects locations updated during time 0.5 Δt_{mu} to time Δt_{mu} are indexed in partition 3 (as indicated by arrows). As time elapses, repeatedly the first range expires (shaded area), and a new range is appended (dashed line). This use of rolling ranges enables the B^x-tree to handle time effectively.



Indexing of Moving Objects, B^x-Tree, Figure 1 An example of the B^x-tree

Historical Background

Traditional indexes for multidimensional databases, such as the R-tree [2] and its variants were, implicitly or explicitly, designed with the main objective of supporting efficient query processing as opposed to enabling efficient updates. This works well in applications where queries are relatively much more frequent than updates. However, applications involving the indexing of moving objects exhibit workloads characterized by heavy loads of updates in addition to frequent queries.

Several new index structures have been proposed for moving-object indexing. One may distinguish between indexing of the past positions versus indexing of the current and near-future positions of spatial objects. The B^x-tree belongs to the latter category.

Past positions of moving objects are typically approximated by polylines composed of line segments. It is possible to index line segments by R-trees, but the trajectory memberships of segments are not taken into account. In contrast to this, the spatio-temporal R-tree [3] attempts to also group segments according to their trajectory memberships, while also taking spatial locations into account. The trajectory-bundle tree [3] aims only for trajectory preservation, leaving other spatial properties aside. Another example of this category is the multi-version 3DR-tree [4], which combines multi-version B-trees and R-trees. Using partial persistence, multi-version B-trees guarantee time slice performance that is independent of the length of the history indexed.

The representations of the current and near-future positions of moving objects are quite different, as are the indexing challenges and solutions. Positions are represented as points (constant functions) or functions of time, typically linear functions. The Lazy Update R-tree [5] aims to reduce update cost by handling updates of objects that do not move outside their leaf-level MBRs specially, and a generalized approach to bottom-up update in R-trees has recently been examined [6].

Tayeb et al. [7] use PMR-quadtrees for indexing the future linear trajectories of one-dimensional moving points as line segments in (x, t)-space. The segments span the time interval that starts at the current time and extends some time into the future, after which time, a new tree must be built. Kollis et al. [8] employ dual transformation techniques which represent the position of an object moving in a d-dimensional space as a point in a $2d$-dimensional space. Their work is largely theoretical in nature. Based on a similar technique, Patel et al. [9] have most recently developed a practical indexing method, termed STRIPES, that supports efficient updates and queries at the cost of higher space requirements. Another representative indexes are the TPR-tree (time-parameterized R-tree) family of indexes (e. g., [10, 11]), which add the time parameters to bounding boxes in the traditional R-tree.

Scientific Fundamentals

Index Structure

The base structure of the B^x-tree is that of the B^+-tree. Thus, the internal nodes serve as a directory. Each internal node contains a pointer to its right sibling (the pointer is non-null if one exists). The leaf nodes contain the moving-object locations being indexed and corresponding index time.

To construct the B^x-tree, the key step is to map object locations to single-dimensional values. A space-filling curve is used for this purpose. Such a curve is a continuous path which visits every point in a discrete, multi-dimensional space exactly once and never crosses itself. These curves are effective in preserving proximity, meaning that points close in multidimensional space tend to be close in the one-dimensional space obtained by the curve. Current versions of the B^x-tree use the Peano curve (or Z-curve) and the Hilbert curve. Although other curves may be used, these two are expected to be particularly good according to analytical and empirical studies in [12]. In what follows, the value obtained from the space-filling curve is termed as the *x_value*.

An object location is given by $O = (\vec{x}, \vec{v})$, a position and a velocity, and an update time, or timestamp, t_u, where these values are valid. Note that the use of linear functions reduces the amount of updates to one third in comparison to constant functions. In a leaf-node entry, an object O updated at t_u is represented by a value $B^x value(O, t_u)$:

$$B^x value(O, t_u) = [index\_partition]_2 \oplus [x\_rep]_2 \quad (1)$$

where *index_partition* is an index partition determined by the update time, *x_rep* is obtained using a space-filling curve, $[x]_2$ denotes the binary value of x, and $\oplus$ denotes concatenation.

If the timestamped object locations are indexed without differentiating them based on their timestamps, the proximity preserving property of the space-filling curve will be lost; and the index will also be ineffective in locating an object based on its *x_value*. To overcome such problems, the index is "partitioned" by placing entries in partitions based on their update time. More specifically, Δt_{mu} denotes the time duration that is the maximum duration in-between two updates of any object location. Then the time axis is partitioned into intervals of duration Δt_{mu}, and each such interval is sub-partitioned into n equal-length subintervals, termed *phases*. By mapping the update times in the same phase to the same so-called *label timestamp* and by using the label timestamps as prefixes of the representations of the object locations, index partitions are obtained, and the update times of updates determine the partitions they go to. In particular, an update with timestamp t_u is assigned a label timestamp $t_{lab} = \lceil t_u + \Delta t_{mu}/n \rceil_l$, where operation $\lceil x \rceil_l$ returns the nearest future label timestamp of x. For example, Fig. 1 shows a B^x-tree with $n=2$. Objects with timestamp $t_u = 0$ obtain label timestamp $t_{lab} = 0.5\,\Delta t_{mu}$; objects with $0 < t_u \leq 0.5\Delta t_{mu}$ obtain label timestamp $t_{lab} = \Delta t_{mu}$; and so on. Next, for an object with label timestamp t_{lab}, its position at t_{lab} is computed according to its position and velocity at t_u. Then the space-filling curve is applied to this (future) position to obtain the second component of Eq. 1.

This mapping has two main advantages. First, it enables the tree to index object positions valid at different times, overcoming the limitation of the B^+-tree, which is only able to index a snapshot of all positions at the same time. Second, it reduces the update frequency compared to having to update the positions of all objects at each timestamp when only some of them need to be updated. The two components of the mapping function in Eq. 1 are consequently defined as follows:

$$index\_partition = (t_{lab}/(\Delta t_{mu}/n) - 1)\, mod(n+1)$$

$$x\_rep = x\_value(\overrightarrow{x} + \overrightarrow{v} \cdot (t_{lab} - t_u))$$

With the transformation, the B^x-tree will contain data belonging to $n+1$ phases, each given by a *label timestamp* and corresponding to a time interval. The value of n needs to be carefully chosen since it affects query performance and storage space. A large n results in smaller enlargements of query windows (covered in the following subsection), but also results in more partitions and therefore a looser relationship among object locations. In addition, a large n yields a higher space overhead due to more internal nodes.

To exemplify, let $n=2$, $\Delta t_{mu} = 120$, and assume a Peano curve of order 3 (i. e., the space domain is 8×8). Object positions $O_1 = ((7,2), (-0.1, 0.05))$, $O_2 = ((0,6), (0.2, -0.3))$, and $O_3 = ((1,2), (0.1, 0.1))$ are inserted at times 0, 10, and 100, respectively. The $B^x value$ for each object is calculated as follows.

Step 1: Calculate label timestamps and index partitions.

$$t^1_{lab} = \lceil (0 + 120/2) \rceil_l = 60,$$
$$index\_partition^1 = 0 = (00)_2$$
$$t^2_{lab} = \lceil (10 + 120/2) \rceil_l = 120,$$
$$index\_partition^2 = 1 = (01)_2$$
$$t^3_{lab} = \lceil (100 + 120/2) \rceil_l = 180,$$
$$index\_partition^3 = 2 = (10)_2$$

Step 2: Calculate positions x_1, x_2, and x_3 at t^1_{lab}, t^2_{lab}, andt^3_{lab}, respectively.

$$x'_1 = (1, 5)$$
$$x'_2 = (2, 3)$$
$$x'_3 = (4, 1)$$

Step 3: Calculate Z-values.

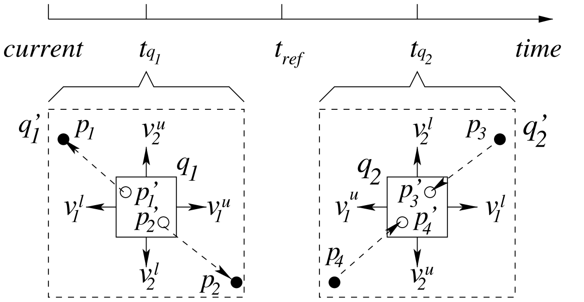
$$[Z\_value(x'_1)]_2 = (010011)_2$$
$$[Z\_value(x'_2)]_2 = (001101)_2$$
$$[Z\_value(x'_3)]_2 = (100001)_2$$

Range Query Algorithm

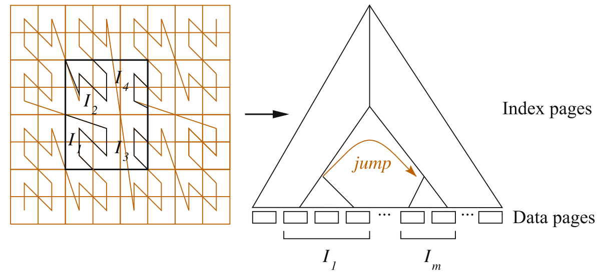
A range query retrieves all objects whose location falls within the rectangular range $q = ([qx^l_1, qx^u_1], [qx^l_2, qx^u_2])$ at time t_q not prior to the current time ("l" denotes lower bound, and "u" denotes upper bound).

A key challenge is to support predictive queries, i. e., queries that concern future times. Traditionally, indexes that use linear functions handle predictive queries by means of bounding box enlargement (e. g., the TPR-tree). Whereas, the B^x-tree uses query-window enlargement. Since the B^x-tree stores an object's location as of some time after its update time, the enlargement involves two cases: a location must either be brought back to an earlier time or forward to a later time. Consider the example in Fig. 2, where t_{ref} denotes the time when the locations of four moving objects are updated to their current value index, and where predictive queries q_1 and q_2 (solid rectangles) have time parameters t_{q1} and t_{q2}, respectively.

The figure shows the stored positions as solid dots and positions of the two first objects at t_{q1} and the positions



Indexing of Moving Objects, B$^{\mathrm{x}}$-Tree, Figure 2 Query window enlargement



Indexing of Moving Objects, B$^{\mathrm{x}}$-Tree, Figure 3 Jump in the index

of the two last at t_{q2} as circles. The two positions for each object are connected by an arrow. The relationship between the two positions for each object is $p_i' = p_i + \vec{v} \cdot (t_q - t_{ref})$. The first two of the four objects, thus, are in the result of the first query, and the last two objects are in the result of the second query. To obtain this result, query rectangle q_1 needs to be enlarged to q_1' (dashed). This is achieved by attaching maximum speeds to the sides of $q_1 : v_1^l, v_2^l, v_1^u,$ andv_2^u. For example, v_1^u is obtained as the largest projection onto the x-axis of a velocity of an object in q_1'. For q_2, the enlargement speeds are computed similarly. For example, v_2^u is obtained by projecting all velocities of objects in q_2' onto the y-axis; v_2^u is then set to the largest speed multiplied by –1.

The enlargement of query $q = ([qx_1^l, qx_1^u], [qx_2^l, qx_2^u])$ is given by query $q' = ([eqx_1^l, eqx_1^u], [eqx_2^l, eqx_2^u])$:

$$eqx_i^l = \begin{cases} qx_i^l + v_i^l \cdot (t_{ref} - t_q) & \text{if } t_q < t_{ref} \\ qx_i^l + v_i^u \cdot (t_q - t_{ref}) & \text{otherwise} \end{cases} \quad (2)$$

$$eqx_i^u = \begin{cases} qx_i^u + v_i^u \cdot (t_{ref} - t_q) & \text{if } t_q < t_{ref} \\ qx_i^u + v_i^l \cdot (t_q - t_{ref}) & \text{otherwise} \end{cases} . \quad (3)$$

The implementation of the computation of enlargement speeds proceeds in two steps. They are first set to the maximum speeds of all objects, thus a preliminary q' is obtained. Then, with the aid of a two-dimensional histogram (e. g., a grid) that captures the maximum and minimum projections of velocities onto the axes of objects in each cell, the final enlargement speed in the area where the query window resides is obtained. Such a histogram can easily be maintained in main memory.

Next, the partitions of the B$^{\mathrm{x}}$-tree are traversed to find objects falling in the enlarged query window q'. In each partition, the use of a space-filling curve means that a range query in the native, two-dimensional space becomes a set of range queries in the transformed, one-dimensional space (see Fig. 3); hence multiple traversals of the index result. These traversals are optimized by calculating the start and end points of the one-dimensional ranges and traverse the intervals by "jumping" in the index.

k Nearest Neighbor Query Algorithm

Assuming a set of $N>k$ objects and given a query object with position $q = (qx_1, qx_2)$, the k nearest neighbor query (kNN query) retrieves k objects for which no other objects are nearer to the query object at time t_q not prior to the current time.

This query is computed by iteratively performing range queries with an incrementally enlarged search region until k answers are obtained. First, a range R_{q1} centered at q with extension $r_q = D_k/k$ is constructed. D_k is the estimated distance between the query object and its k'th nearest neighbor; D_k can be estimated by the following equation [13]:

$$D_k = \frac{2}{\sqrt{\pi}} \left[1 - \sqrt{1 - \left(\frac{k}{N}\right)^{1/2}} \right] .$$

The range query with range R_{q1} at time t_q is computed, by enlarging it to a range R_{q1}' and proceeding as described in the previous section. If at least k objects are currently covered by R_{q1}' and are enclosed in the inscribed circle of R_{q1} at time t_q, the kNN algorithm returns the k nearest objects and then stops. It is safe to stop because all the objects that can possibly be in the result have been considered. Otherwise, R_{q1} is extended by r_q to obtain R_{q2} and an enlarged window R_{q2}'. This time, the region $R_{q2}' - R_{q1}'$ is searched and the neighbor list is adjusted accordingly. This process is repeated until we obtain an R_{qi} so that there are k objects within its inscribed circle.

Continuous Query Algorithm

The queries considered so far in this section may be considered as one-time queries: they run once and complete when a result has been returned. Intuitively, a continuous query is a one-time query that is run at each point in time during a time interval. Further, a continuous query takes a *now*-relative time $now + \Delta t_q$ as a parameter instead of the fixed time t_q. The query then maintains the result of the corresponding one-time query at time $now + \Delta t_q$ from when the query is issued at time t_{issue} and until it is deactivated.

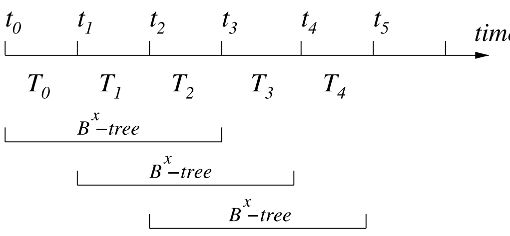
Such a query can be supported by a query q_e with time interval $[t_{issue} + \Delta t_q, t_{issue} + \Delta t_q + l]$ ("l" is a time interval) [14]. Query q_e can be computed by the algorithms presented previously, with relatively minor modifications: (i) use the end time of the time interval to perform forward enlargements, and use the start time of the time interval for backward enlargements; (ii) store the answer sets during the time interval. Then, from time t_{issue} to $t_{\text{issue}} + l$, the answer to q_l is maintained during update operations. At $t_{\text{issue}} + l$, a new query with time interval $[t_{issue} + \Delta t_q + l, t_{issue} + \Delta t_q + 2l]$ is computed.

A continuous range query during updates can be maintained by adding or removing the object from the answer set if the inserted or deleted object resides in the query window. Such operations only introduce CPU cost.

The maintenance of continuous *k*NN queries is somewhat more complex. Insertions also only introduce CPU cost: an inserted object is compared with the current answer set. Deletions of objects not in the answer set does not affect the query. However, if a deleted object is in the current answer set, the answer set is no longer valid. In this case, a new query with a time interval of length l at the time of the deletion is issued. If the deletion time is t_{del}, a query with time interval $[t_{del} + \Delta t_q, t_{del} + \Delta t_q + l]$ is triggered at t_{del}, and the answer set is maintained from t_{del} to $t_{del} + l$. The choice of the "optimal" l value involves a trade-off between the cost of the computation of the query with the time interval and the cost of maintaining its result. On the one hand, a small l needs to be avoided as this entails frequent recomputations of queries, which involve a substantial I/O cost. On the other hand, a large l introduces a substantial cost: Although computing one or a few queries is cost effective in itself, the cost of maintaining the larger answer set must also be taken into account, which may generate additional I/Os on each update. Note that maintenance of continuous range queries incur only CPU cost. Thus, a range query with a relatively large l is computed such that l is bounded by $\Delta t_{mu} - \Delta t_q$ since the answer set obtained at t_{issue} is no longer valid at $t_{\text{issue}} + \Delta t_{mu}$. For the continuous *k*NN queries, l needs to be carefully chosen.

Update, Insertion, and Deletion

Given a new object, its index key is calculated according to Eq. 1, and then insert it into the B^x-tree as in the B$^+$-tree. To delete an object, an assumption is made that the positional information for the object used at its last insertion and the last insertion time are known. Then its index key is calculated and the same deletion algorithm as in the B$^+$-tree is employed. Therefore, the B^x-tree directly inherits the good properties of the B$^+$-tree, and efficient update performance is expected.



Indexing of Moving Objects, B^x-Tree, Figure 4 B^x-tree evolution

However, one should note that update in the B^x-tree does differ with respect to update in the B$^+$-tree. The B^x-tree only updates objects when their moving functions have been changed. This is realized by clustering updates during a certain period to one time point and maintaining several corresponding sub-trees. The total size of the three subtrees is equal to that of one tree indexing all the objects.

In some applications, there may be some object positions that are updated relatively rarely. For example, most objects may be updated at least each 10 minutes, but a few objects are updated once a day. Instead of letting outliers force a large maximum update interval, a "maximum update interval" within which a high percentage of objects have been updated is used. Object positions that are not updated within this interval are "flushed" to a new partition using their positions at the label timestamp of the new partition. In the example shown in Fig. 4, suppose that some object positions in T_0 are not updated at the time when T_0 expires. At this time, these objects are moved to T_2. Although this introduces additional update cost, the (controllable) amortized cost is expected to be very small since outliers are rare. The forced movement of an object's position to a new partition does not cause any problem with respect to locating the object, since the new partition can be calculated based on the original update time. Likewise, the query efficiency is not affected.

Key Applications

With the advances in positioning technologies, such as GPS, and rapid developments of wireless communication devices, it is now possible to track continuously moving objects, such as vehicles, users of wireless devices and goods. The B^x-tree can be used in a number of emerging applications involving the monitoring and querying of large quantities of continuous variables, e. g., the positions of moving objects. In the following, some of these applications are discussed.

Location-Based Service

A traveller comes to a city that he is not familiar with. To start with, he sends his location by using his PDA or smart phone (equipped with GPS) to a local server that provides location-based services. Then the service provider can answer queries like "where is the nearest restaurant (or hotel)?" and can also help to dispatch a nearby taxi to the traveller.
A driver can also benefit from the location-based services. For example, he can ask for a nearest gas station or motel when he is driving.

Traffic Control

If the moving objects database stores information about locations of vehicles, it may be able to predict the possible congestion in near future. To avoid the congestion, the system can divert some vehicles to alternate routes in advance.
For air traffic control, the moving objects database system can retrieve all the aircrafts within a certain region and prevent a possible collision.

E-commerce

In these applications, stores send out advertisements or e-coupons to vehicles passing by or within the store region.

Digital Game

Another interesting example is location-based digital game where the positions of the mobile users play a central role. In such games, players need to locate their nearest neighbors to fulfill "tasks" such as "shooting" other close players via their mobile devices.

Battle Field

The moving object database technique is also very important in the military. With the help of the moving object database techniques, helicopters and tanks in the battlefield may be better positioned and mobilized to the maximum advantage.

RFID Application

Recently, applications using radio frequency identification (RFID) has received much interest. RFID enables data to be captured remotely via radio waves and stored on electronic tags embedded in their carriers. A reader (scanner) is then used to retrieve the information. In a hospital application, RFIDs are tagged to all patients, nurses and doctors, so that the system can keep a real-time tracking of their movements. If there is an emergency, nurses and doctors can be sent to the patients more quickly.

Future Directions

Several promising directions for future work exist. One could be the improvement of the range query performance in the B^x-tree since the current range query algorithm uses the strategy of enlarging query windows which may incur some redundant search. Also, the use of the B^x-tree for the processing of new kinds of queries can be considered. Another direction is the use of the B^x-tree for other continuous variables than the positions of mobile service users. Yet another direction is to apply the linearization technique to other index structures.

Cross References

► Indexing, BDual Tree
► Indexing the Positions of Continuously Moving Objects

Recommended Reading

1. Jensen, C.S., Lin, D., Ooi, B.C.: Query and update efficient B+-tree based indexing of moving objects. In: Proc. VLDB, pp. 768–779 (2004)
2. Guttman, A.: R-trees: A Dynamic Index Structure for Spatial Searching. In: Proc. ACM SIGMOD, pp. 47–57 (1984)
3. Pfoser, D., Jensen, C.S., Theodoridis, Y.: Novel approaches in query processing for moving objects. In: Proc. VLDB, pp. 395–406 (2000)
4. Tao, Y., Papadias, D.: MV3R-tree: a spatio-temporal access method for timestamp and interval queries. In: Proc. VLDB, pp. 431–440 (2001)
5. Kwon, D., Lee, S., Lee, S.: Indexing the current positions of moving objects using the lazy update R-tree. In: Proc. MDM, pp. 113–120 (2002)
6. Lee, M.L., Hsu, W., Jensen, C.S., Cui, B., Teo, K.L.: Supporting frequent updates in R-trees: a bottom-up approach. In: Proc. VLDB, pp. 608–619 (2003)
7. Tayeb, J., Ulusoy, O., Wolfson, O.: A quadtree based dynamic attribute indexing method. Computer J. **41**(3):185–200 (1998)
8. Kollios, G., Gunopulos, D., Tsotras, V.J.: On indexing mobile objects. In: Proc. PODS, pp. 261–272 (1999)
9. Patel, J.M., Chen Y., Chakka V.P.: STRIPES: An efficient index for predicted trajectories. In: Proc. ACM SIGMOD, (2004) (to appear)
10. Saltenis, S., Jensen, C.S., Leutenegger, S.T., Lopez, M.A.: Indexing the positions of continuously moving objects. In: Proc. ACM SIGMOD, pp. 331–342 (2000)
11. Tao, Y., Papadias, D., Sun, J.: The TPR*-tree: an optimized spatio-temporal access method for predictive queries. In: Proc. VLDB, pp. 790–801 (2003)

12. Moon, B., Jagadish, H.V., Faloutsos, C., Saltz J.H.: Analysis of the clustering properties of the hilbert space-filling curve. IEEE TKDE **13**(1):124–141 (2001)
13. Tao, Y., Zhang, J., Papadias, D., Mamoulis, N.: An efficient cost model for optimization of nearest neighbor search in low and medium dimensional spaces. IEEE TKDE **16**(10):1169–1184 (2004)
14. Benetis, R., Jensen, C.S., Karciauskas, G., Saltenis, S.: Nearest neighbor and reverse nearest neighbor queries for moving objects. In: Proc. IDEAS, pp. 44–53 (2002)

Indexing, Parametric Space

► Indexing Spatio-temporal Archives
► Movement Patterns in Spatio-temporal Data

Indexing, Query and Velocity-Constrained

SUNIL PRABHAKAR, DMITRI KALASHNIKOV, YUNI XIA
Department of Computer Sciences, Purdue University, West Lafayette, IN, USA

Synonyms

Spatio-temporal indexing; Continuous queries

Definition

Moving object environments are characterized by large numbers of moving objects and numerous concurrent continuous queries over these objects. Efficient evaluation of these queries in response to the movement of the objects is critical for supporting acceptable response times. In such environments the traditional approach of building an index on the objects (data) suffers from the need for frequent updates and thereby results in poor performance. In fact, a brute force, no-index strategy yields better performance in many cases. Neither the traditional approach, nor the brute force strategy achieve reasonable query processing times. The efficient and scalable evaluation of multiple continuous queries on moving objects can be achieved by leveraging two complimentary techniques: *Query Indexing* and *Velocity Constrained Indexing* (*VCI*). Query Indexing relies on i) incremental evaluation; ii) reversing the role of queries and data; and iii) exploiting the relative locations of objects and queries. VCI takes advantage of the maximum possible speed of objects in order to delay the expensive operation of updating an index to reflect the movement of objects. In contrast to techniques that require exact knowledge about the movement of the objects, VCI does not rely on such information. While Query Indexing outperforms VCI, it does not efficiently handle the arrival of new queries. Velocity constrained indexing, on the other hand, is unaffected by changes in queries. A combination of Query Indexing and Velocity Constrained Indexing enables the scalable execution of insertion and deletion of queries in addition to processing ongoing queries.

Historical Background

The importance of moving object environments is reflected in the significant body of work addressing issues such as indexing, uncertainty management, broadcasting, and models for spatio-temporal data. Several indexing techniques for moving objects have been proposed. These include indexes over the histories, or trajectories, of the positions of moving objects, or the current and anticipated future positions of the moving objects. Uncertainty in the positions of the objects is dealt with by controlling the update frequency where objects report their positions and velocity vectors when their actual positions deviate from what they have previously reported by some threshold. Tayeb et al. use quad-trees to index the trajectories of one-dimensional moving points. In one approach, moving objects and their velocities are mapped to points which are indexed using a kD-tree. Another indexes the past trajectories of moving objects treated as connected line segments. Yet another considers the management of collections of moving points in the plane by describing the current and expected positions of each point in the future [3]. They address how often to update the locations of the points to balance the costs of updates against imprecision in the point positions. The two techniques presented here appeared in [2].

Scientific Fundamentals

Continuous Query Processing

Location-based environments are characterized by large numbers of moving (and stationary) objects. To support these services it is necessary to execute efficiently several types of queries, including range queries, nearest-neighbor queries, density queries, etc. An important requirement in location-aware environments is the continuous evaluation of queries. Given the large numbers of queries and moving objects in such environments, and the need for a timely response for continuous queries, efficient and scalable query execution is paramount.

This discussion focuses on range queries. Range queries arise naturally in spatial applications such as a query that needs to keep track of, for example, the number of people that have entered a building. Range queries can also be useful as pre-processing tools for reducing the amount of data that other queries, such as nearest-neighbor or density, need to process.

Model

Moving objects are represented as points, and queries are expressed as rectangular spatial regions. Therefore, given a collection of moving objects and a set of queries, the problem is to identify which objects lie within (i. e., are relevant to) which queries. Objects report their new locations periodically or when they have moved by a significant distance. Updates from different objects arrive continuously and asynchronously. The location of each object is saved in a file. Objects are required to report only their location, not velocity. There is no constraint on the movement of objects except that the maximum possible speed of each object is known and not exceeded (this is required only for Velocity Constrained Indexing).

Ideally, each query should be re-evaluated as soon as an object moves. However, this is impractical and may not even be necessary from the user's point of view. The continuous evaluation of queries takes place in a periodic fashion whereby the set of objects that are relevant to each continuous query are determined at fixed time intervals.

Limitations of Traditional Indexing

A natural choice for efficient evaluation of range queries is to build a spatial index on the objects. To determine which objects intersect each query, the queries are executed using this index. The use of the spatial index should avoid many unnecessary comparisons of queries against objects this approach should outperform the brute force approach. This is in agreement with conventional wisdom on indexing. In order to evaluate the answers correctly, it is necessary to keep the index updated with the latest positions of objects as they move. This represents a significant problem. Notice that for the purpose of evaluating continuous queries, only the current snapshot is relevant and not the historical movement of objects.

In [2], three alternatives for keeping an index updated are evaluated. Each of these gives very poor performance – even worse than the naive brute-force approach. The poor performance of the traditional approach of building an index on the data (i. e. the objects) can be traced to the following two problems: i) whenever *any* object moves, it becomes necessary to re-execute *all* queries; and ii) the cost of keeping the index updated is very high.

Query Indexing

The traditional approach of using an index on object locations to efficiently process queries for moving objects suffers from the need for constant updates to the index and re-evaluation of all queries whenever any object moves. These problems can be overcome by employing two key ideas:

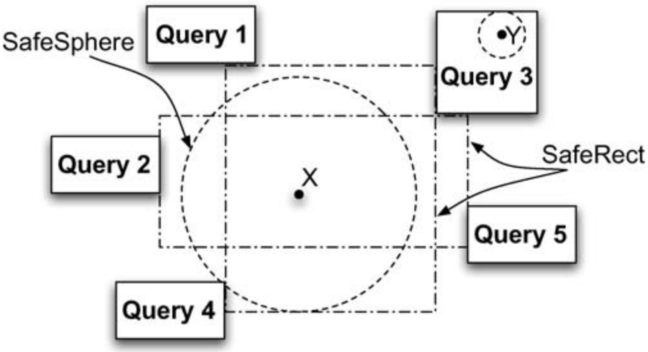
- *reversing* the role of data and queries, and
- *incremental* evaluation of continuous queries.

The notion of *safe regions* that exploit the relative location of objects and queries can further improve performance.

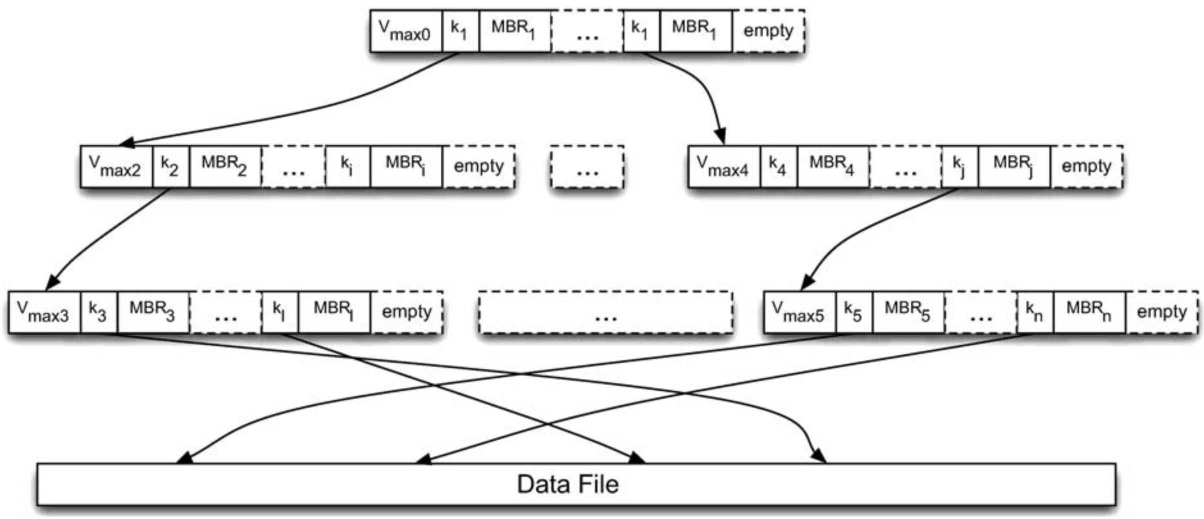
In treating the queries as data, a spatial index such as an R-tree is built on the queries instead of the customary index that is built on the objects (i. e. data). This structure is called a Query-Index or *Q-index*. To evaluate the intersection of objects and queries, each object is treated as a "query" on the Q-index (i. e., the moving objects are treated as queries in the traditional sense). Exchanging queries for data results in a situation where a larger number of queries (one for each object) is executed on a smaller index (the Q-index), as compared to an index on the objects. This is not necessarily advantageous by itself. However, since not all objects change their location at each time step, a large number of "queries" on the Q-index can be avoided by incrementally maintaining the result of the intersection of objects and queries.

Incremental evaluation is achieved as follows: upon creation of the Q-index, all objects are processed on the Q-index to determine the initial result. Following this, the query results are incrementally adjusted by considering the movement of objects. At each evaluation time step, only those objects that have moved since the last time step are processed, and adjust their relevance to queries accordingly. If most objects do not move during each time step, this can greatly reduce the number of times the Q-index is accessed. For objects that move, the Q-index improves the search performance as compared to a comparison against all queries.

Under traditional indexing, at each time step, it would be necessary to update the index on the objects and then evaluate each query on the modified index. This is independent of the movement of objects. With the "Queries as Data" or the Q-index approach, only the objects that have moved since the previous time step are evaluated against the Q-index. Building an index on the queries avoids the high cost of keeping an object index updated; incremental evaluation exploits the smaller numbers of objects that move in a single time step to avoid repeating unnecessary comparisons. Upon the arrival of a new query, it is necessary to compare the query with all the objects in order to initiate the incremental processing. Deletion of queries is easily handled by ignoring those queries. Further improvements in performance can be achieved by taking into account the relative locations of objects and queries or safe regions.



Indexing, Query and Velocity-Constrained, Figure 1 Examples of *Safe Regions*



Indexing, Query and Velocity-Constrained, Figure 2 Example of Velocity Constrained Index (VCI)

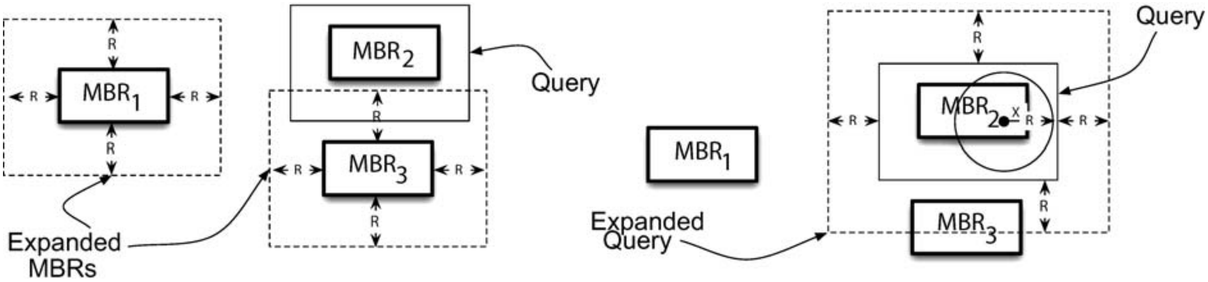
***Safe Regions*: Exploiting Query and Object Locations** The relevance of an object with respect to a given query can only change if the object crosses a query boundary. Therefore, a region surrounding an object that does not cross any query boundary represents a region within which the object can move without affecting the result of any query. Such a region is termed a *Safe Region*. Two types of safe regions are maximal circles (sphere in general) and maximal rectangles centered at an object's current location. These are termed *SafeSphere* and *SafeRect* respectively. While there is only one maximal sphere for a given object, there can be multiple maximal rectangles.

Figure 1 shows examples of each type of *Safe Region* for two object locations, X and Y. Note that the union of two safe regions is also a safe region. If an object knows a safe region around it, it need not send updates of its movement to the server as long as it remains within the safe region. The safe region optimizations significantly reduce the need to test data points for relevance to queries if they are far from any query boundaries and move slowly. Using safe regions can significantly improve performance.

Velocity-Constrained Indexing

The technique of Velocity-Constrained Indexing (VCI) eliminates the need for continuous updates to an index on moving objects by relying on the notion of a maximum speed for each object. Under this model, the maximum possible speed of each object is known.

A VCI is a regular R-tree based index on moving objects with an additional field in each node: v_{max}. This field stores the maximum allowed speed over all objects covered by that node in the index. The v_{max} entry for an internal node is simply the maximum of the v_{max} entries of its children. The v_{max} entry for a leaf node is the maximum allowed speed among the objects pointed to by the node. Figure 2 shows an example of a VCI. The v_{max} entry in each node is maintained in a manner similar to the Minimum Bounding



Indexing, Query and Velocity-Constrained, Figure 3 Query Processing with Velocity Constrained Index (VCI)

Rectangle (MBR) of each entry in the node, except that there is only one v_{max} entry per node as compared to an MBR per entry of the node. When a node is split, the v_{max} for each of the new nodes is copied from the original node. Consider a VCI that is constructed at time t_0. At this time it accurately reflects the locations of all objects. At a later time t, the same index does not accurately capture the correct locations of points since they may have moved arbitrarily. Normally the index needs to be updated to be correct. However, the v_{max} fields enable us to use this old index without updating it. We can safely assert that no point will have moved by a distance larger than $R = v_{max}(t - t_0)$. If each MBR is expanded by this amount in all directions, the expanded MBRs will correctly enclose all underlying objects. Therefore, in order to process a query at time t, the VCI created at time t_0 can be used without being updated, by simply comparing the query with expanded version of the MBRs saved in VCI. At the leaf level, each point object is replaced by a square region of side $2R$ for comparison with the query rectangle.[1]

An example of the use of the VCI is shown in Fig. 3a which shows how each of the MBRs in the same index node are expanded and compared with the query. The expanded MBR captures the worst-case possibility that an object that was at the boundary of the MBR at t_0 has moved out of the MBR region by the largest possible distance. Since a single v_{max} value is stored for all entries in the node, each MBR is expanded by the same distance, $R = v_{max}(t - t_0)$. If the expanded MBR intersects with the query, the corresponding child is searched. Thus to process a node all the MBRs stored in the node (except those that intersect without expansion) need to be expanded. Alternatively, a single expansion of the query by R could be performed and compared with the unexpanded MBRs. An MBR will intersect with the expanded query if and only if the same MBR after expansion intersects with the original query. Figure 3b shows the earlier example with query expansion. Expanding the query once per node saves some unnecessary computation.

The set of objects found to be in the range of the query based upon an old VCI is a superset, S', of the exact set of objects that are currently in the query's range. Clearly, there can be no false dismissals in this approach. In order to eliminate the false positives, it is necessary to determine the current positions of all objects in S'. This is achieved through a post-processing step. The current location of the object is retrieved from disk and compared with the query to determine the current matching. Note that it is not always necessary to determine the current location of each object that falls within the expanded query. From the position recorded in the leaf entry for an object, it can move by at most R. Thus its current location may be anywhere within a circle of radius R centered at the position recorded in the leaf. If this circle is entirely contained within the unexpanded query, there is no need to post-process this object for that query. Object X in Fig. 3b is an example of such a point.

To avoid performing an I/O operation for each object that matches each expanded query, it is important to handle the post-processing carefully. We can begin by first preprocessing all the queries on the index to identify the set of objects that need to be retrieved for any query. These objects are then retrieved only once and checked against all queries. This eliminates the need to retrieve the same object more than once. To avoid multiple retrievals of a page, the objects to be retrieved can first be sorted on page number. Alternatively, a clustered index can be built. Clustering may reduce the total number of pages to be retrieved. Clustering the index can improve the performance significantly.

Refresh and Rebuild

The amount of expansion needed during query evaluation depends upon two factors: the maximum speed v_{max} of the node, and the time that has elapsed since the index

[1] Note that it should actually be replaced by a circle, but the rectangle is easier to handle.

was created, $(t-t_0)$. Thus over time the MBRs get larger, encompassing more and more dead space, and may not be minimal. Consequently, as the index gets older its quality gets poorer. Therefore, it is necessary to *rebuild* the index periodically. This essentially resets the creation time, and generates an index reflecting the changed positions of the objects. Rebuilding is an expensive operation and cannot be performed too often. A cheaper alternative to rebuilding the index is to *refresh* it. Refreshing simply updates the locations of objects to the current values and adjusts the MBRs so that they are minimal. Following refresh, the index can be treated as though it has been rebuilt.

Refreshing can be achieved efficiently by performing a depth-first traversal of the index. For each entry in a leaf node the latest location of the object is retrieved (sequential I/O if the index is clustered). The new location is recorded in the leaf page entry. When all the entries in a leaf node are updated, the MBR for the node is computed and recorded it in the parent node. For directory nodes when all MBRs of its children have been adjusted, the overall MBR for the node is computed and recorded it in the parent. This is very efficient with depth-first traversal. Although refresh is more efficient than a rebuild, it suffers from not altering the structure of the index – it retains the earlier structure. If points have moved significantly, they may better fit under other nodes in the index. Thus there is a trade-off between the speed of refresh and the quality of the index. An effective solution is to apply several refreshes followed by a less frequent rebuild. In practice, refreshing works very well thereby avoiding the need for frequent, expensive rebuilds.

Performance

Detailed evaluation of the approaches can be found in [2]. Overall, the experiments show that for Query Indexing, *SafeRect* is the most effective in reducing evaluations. Q-Index is found to give the best performance compared to VCI, traditional indexing, and sequential scans. It is also robust across various scales (numbers of objects, queries), rates of movement, and density of objects versus queries. VCI, on the other hand, is more effective than traditional approaches for small numbers of queries. Moreover, the total cost of VCI approaches that of a sequential scan after some time. Clustering can extend the utility of the VCI index for a longer period of time. Eventually, refresh or rebuilds are necessary. Refreshes are much faster and very effective and thereby reduce the need for frequent rebuilds. VCI is not affected by how often objects move (unlike Q-Index). Thus the costs would not change even if all the objects were moving at each time instant. On the other hand, the VCI approach is very sensitive to the number of queries.

Combined Indexing Scheme

The results show that query indexing and safe region optimizations significantly outperform the traditional indexing approaches and also the VCI approach. These improvements in performance are achieved by eliminating the need to evaluate all objects at each time step through incremental evaluation. Thus they perform well when there is little change in the queries being evaluated. The deletion of queries can be easily handled simply by ignoring the deletion until the query can be removed from the Q-index. The deleted query may be unnecessarily reducing the safe region for some objects, but this does not lead to incorrect processing and the correct safe regions can be recomputed in a lazy manner without a significant impact on the overall costs.

The arrival of new queries, however, is expensive under the query indexing approach as each new query must initially be compared to every object. Therefore a sequential scan of the entire object file is needed at each time step that a new query is received. Furthermore, a new query potentially invalidates the safe regions rendering the optimizations ineffective until the safe regions are recomputed. The VCI approach, on the other hand, is unaffected by the arrival of new queries (only the total number of queries being processed through VCI is important). Therefore to achieve scalability under the insertion and deletion of queries *combined scheme* works best. Under this scheme, both a Q-Index and a Velocity Constrained Index are maintained. Continuous queries are evaluated incrementally using the Q-index and the *SafeRect* optimization. The Velocity Constrained Index is periodically refreshed, and less periodically rebuilt (e. g. when the refresh is ineffective in reducing the cost). New queries are processed using the VCI. At an appropriate time (e. g. when the number of queries being handled by VCI becomes large) all the queries being processed through VCI are transferred to the Query Index in a single step. As long as not too many new queries arrive at a given time, this solution offers scalable performance that is orders of magnitude better than the traditional approaches.

Key Applications

Mobile electronic devices that are able to connect to the Internet have become common. In addition, to being connected, many devices are also able to determine the geographical location of the device through the use of global positioning systems, and wireless and cellular telephone technologies. This combined ability enables new location-based services, including location and mobile commerce (L- and M-commerce). Current location-aware services allow proximity-based queries including map viewing and navigation, driving directions, searches for hotels and restaurants, and weather and traffic information.

These technologies are the foundation for pervasive location-aware environments and services. Such services have the potential to improve the quality of life by adding location-awareness to virtually all objects of interest such as humans, cars, laptops, eyeglasses, canes, desktops, pets, wild animals, bicycles, and buildings. Applications can range from proximity-based queries on non-mobile objects, locating lost or stolen objects, tracing small children, helping the visually challenged to navigate, locate, and identify objects around them, and to automatically annotating objects online in a video or a camera shot. Another example of the importance of location information is the Enhanced 911 (E911) standard. The standard provides wireless users the same level of emergency 911 support as wireline callers.

Future Directions

Natural extensions of these techniques are to support other types of important continuous queries including nearest-neighbor and density queries. Query indexing can easily be extended for these types of queries too. More generally, there are many instances where it is necessary to efficiently maintain an index over data that is rapidly evolving. A primary example is sensor databases. For the applications too, query indexing can be an effective tool.

An important area for future research is the development of index structures that can handle frequent updates to data. Traditionally, index structures have been built with the assumption that updates are not as frequent as queries. Thus, the optimization decisions (for example, the split criteria for R-trees) are made with query performance in mind. However, for high-update environments, these decisions need to be revisited.

Cross References

► Continuous Queries in Spatio-temporal Databases
► Indexing the Positions of Continuously Moving Objects

Recommended Reading

1. Kollios, G., Gunopulos, D., Tsotras, V.J.: On indexing mobile objects. In: Proc. ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), June (1999)
2. Prabhakar, S., Xia, Y., Kalashnikov, D., Aref, W., Hambrusch, S.: Query indexing and velocity constrained indexing: Scalable techniques for continuous queries on moving objects. IEEE Trans. Comput. **51**(10), 1124–1140 (2002)
3. Wolfson, Ouri, Xu, Bo, Chamberlain, Sam and Jiang, L.: Moving objects databases: Issues and solutions. In: Proceedings of the SSDBM Conf. pp. 111–122 (1998)

Indexing, R*-Tree

► R*-tree

Indexing Schemes for Multi-dimensional Moving Objects

KHALED ELBASSIONI[1], AMR ELMASRY[2], IBRAHIM KAMEL[3]
[1] Max-Planck-Institute for Computer Science, Saarbrücken, Germany
[2] Department of Computer Engineering and Systems, Alexandria University, Alexandria, Egypt
[3] University of Sharjah, Sharjah, UAE

I

Synonyms

Indexing moving points; Data-structures; Memory, external; Moving objects; Index lifetime; Disk page; R-tree; TPR-trees; STAR-tree; Dual space-time representation; MB-index

Definition

Indexing schemes are generally data structures that are used to keep track of moving objects whose positions are functions of time. Most of the work in this area is described in the *external memory model*, as the number of mobile objects in the database is assumed to be very large. The objective is to allow for efficient answering of queries that usually involve the current and the predicted future positions of the objects. The main challenges in this case, as compared to the case of static objects, are

(i) *Large number of updates:* in the static case, usually the number of queries is the dominating factor. With moving objects, storing the continuously changing locations directly in the database becomes infeasible, considering the prohibitively large number of updates.
(ii) *Queries about the future are allowed:* this requires that the future positions of the objects be predicted somehow based on the current information and the accuracy of this prediction affects the accuracy of the query results.
(iii) *Frequent optimization overhead:* due to the mobility of the objects, an index optimized to give good query performance on these objects at a certain time might not be good later on. This means that optimization has to be carried out more frequently.

One commonly used solution to (i) is to abstract each object's location as a function of time, which will be stored in the database and update the database only when the function's parameters change. A typical solution to (ii) is to assume that objects are moving with constant speed, and issue an update only when their speeds change. More generally, sampling can be used to obtain the positions of each object at certain points in time, and interpolation/extrapolation can be used to predict the locations of the object in between the sample points or at some time in the future. To solve (iii), indexing mechanisms often employ the notion of an index *lifetime*, which is the time interval during which the index is designed to give good performance. At the end of this period, the index is rebuilt and the optimization procedure is again applied in order to guarantee good performance for the next period.

Scientific Fundamentals

Model

In the most general setting, consider a database $\mathcal{D}$ of N objects moving in d-dimensional space, and assume that the position of object $o \in \mathcal{D}$ at time t is given by $o.\mathbf{y}(t) = (o.y_1(t), \dots, o.y_d(t)) \in \mathbb{R}^d$. Most of the known work assumes that each object moves along a *straight line*, justified by the fact that non-linear functions can be approximated accurately enough by connected straight line segments. In such a case, the position of the object is determined by its initial location $o.\mathbf{a} = o.\mathbf{y}(t_{\text{ref}})$, measured at some reference time t_{ref} (usually the time of last update), and its velocity vector $o.\mathbf{v} = (o.v_1, \dots, o.v_d)$, i. e., $o.\mathbf{y}(t) = o.\mathbf{a} + o.\mathbf{v}(t - t_{\text{ref}})$, for $t \geq t_{\text{ref}}$.

Let B be the disk page size, i. e., the number of units of data that can be processed in a single I/O operation. In the standard external memory model of computation, the efficiency of an algorithm is measured in terms of the number of I/O's required to perform an operation. If N is the number of mobile objects in the database and K is the number of objects reported in a given query, then the minimum number of pages to store the database is $n \stackrel{\text{def}}{=} \lceil \frac{N}{B} \rceil$ and the minimum number of I/O's to report the answer is $k \stackrel{\text{def}}{=} \lceil \frac{K}{B} \rceil$. Thus, the time and space complexity of a given algorithm, under such a model, is measured in terms of parameters n and k.

Typical Queries

The following types of queries have been considered in the literature:

Q_1 *(Window query).* Given an axis-parallel hyper-rectangular query region determined by the two corner points $\mathbf{y}', \mathbf{y}'' \in \mathbb{R}^d$ and two time instants t', t'', report all objects $o \in \mathcal{D}$ which cross the region at some instant between t' and t'', i. e., for which $\mathbf{y}' \leq o.\mathbf{y}(t) \leq \mathbf{y}''$, for some $t' \leq t \leq t''$. When $t' = t''$, the query is sometimes called an *instantaneous* or *timeslice* query.

Q_2 *(Moving query).* Here, the query region is a function of time. For instance, consider a generalization of query Q_1, where the corner points $\mathbf{y}'$, $\mathbf{y}''$ vary linearly with time, i. e., $\mathbf{y}'(t) = \mathbf{y}'_0 + v't$ and $\mathbf{y}''(t) = \mathbf{y}''_0 + v''t$, for some constant vectors $\mathbf{y}'_0, \mathbf{y}''_0, \mathbf{v}', \mathbf{v}'' \in \mathbb{R}^d$. The requirement is to report all objects $o \in \mathcal{D}$ for which $\mathbf{y}'(t) \leq o.\mathbf{y}(t) \leq \mathbf{y}''(t)$, for some time t in a specified interval $[t', t'']$.

Q_3 *(Proximity query).* Given a set of m moving objects $O_1, \dots, O_m$, with motion functions $\mathbf{y}^i(t)$, for $i = 1, \dots, m$, and two time instants t', t'', report all objects $o \in \mathcal{D}$ that will become within a distance δ_i from (i. e., will cross *the neighborhood* of) *some* object O_i, $i = 1, \dots, m$, in the specified time interval, i. e., for which $d(o.\mathbf{y}(t), \mathbf{y}^i(t)) \leq \delta_i$ for *some* $i \in \{1, \dots, m\}$, and some $t' \leq t \leq t''$, where $d(\cdot,\cdot)$ is the Euclidean distance.

Q_4 *(Approximate nearest neighbor query).* Given a point $p \in \mathbb{R}^d$, and a time instant t', report the δ-approximate nearest neighbor of p in $\mathcal{D}$ at time t', i. e., an object $o \in \mathcal{D}$, such that $d(o.\mathbf{y}(t'), p) \leq (1 + \delta) \min_{o' \in \mathcal{D}} d(o'.\mathbf{y}(t'), p)$. When $\delta = 0$, it is required to report the nearest neighbor exactly. One may also ask for the approximate (or exact) k-nearest neighbors of the given point p at t'.

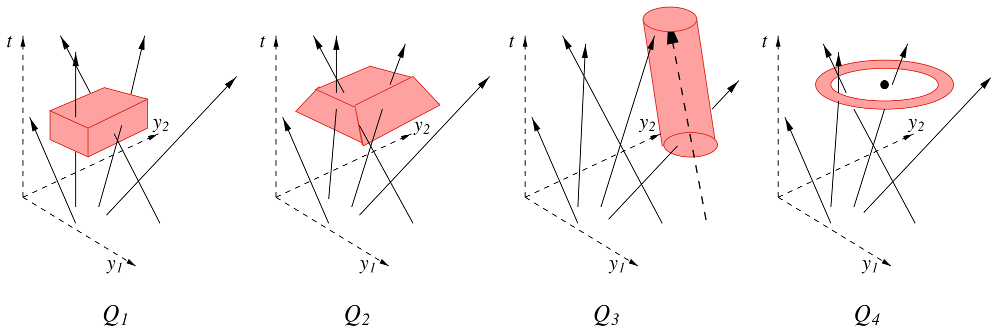
Figure 1 gives examples for these queries. It is assumed that all time instants t', t'' above are at least as large as the current time, i. e., the queries concern the current or future locations of the objects.

Indexing Techniques

Many indices have been recently proposed for indexing moving objects. These techniques can be generally classified into three categories according to the type of moving objects' representation on which they work. Examples of each are given here. Additionally, the reader is referred to [9] and the references therein for a more extensive survey.

1. Explicit Space-Time Representation This is the most straightforward representation. It does not assume linearity of the objects' trajectories. It works by viewing the trajectories of the objects as *static* objects in the $(d+1)$-dimensional space obtained by adding the time as an additional coordinate. Then, any spatial access method, such as quad-trees, R-trees or its variants, can be used to index these trajectories. There is a number of solutions proposed in the literature that belong to this category:

Quad-Tree Based Approach [13]: The 2-dimensional distance-time space (assuming 1-dimensional moving objects) is partitioned into four equal-sized quadrants, each



Indexing Schemes for Multi-dimensional Moving Objects, Figure 1 The four types of queries $Q_1,\ldots, Q_4$

of which is partitioned recursively as in the normal quadtree construction. However, in contrast to the static case, the information about a moving object is stored in every quadrant its trajectory crosses, and this continues recursively until a small number of objects, say at most B, is left in every quadrant. This clearly generalizes to higher dimensions. An obvious drawback is that an object is stored in the index many times, which implies both a large update overhead and a large storage requirement.

R-Tree and Variants A different approach is to approximate each trajectory by its minimum bounding rectangle (MBR), which is then indexed using an *R-tree*, or an R^*-*tree* (see e. g., [2]). However, an MBR assigns a trajectory a much larger area than it really occupies, resulting in what is called *large dead-space*. In addition, the data will be skewed since all trajectories have the same end time value. As a result, these indices usually suffer from excessive overlap between their nodes, eventually leading to poor query performance.

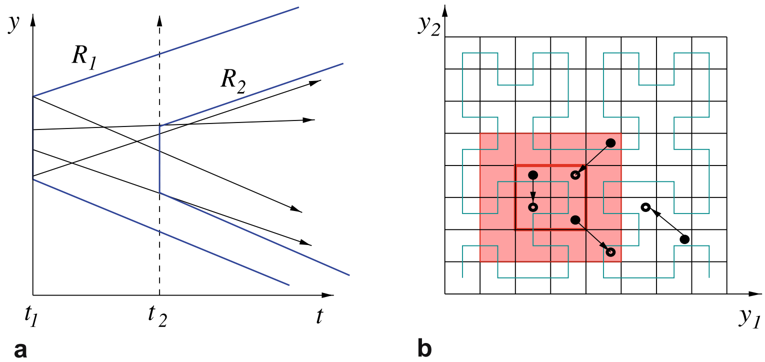
Furthermore, there is a problem in both of the above approaches: at the current time, the trajectories are assumed to extend to infinity. Thus, in order to be able to apply the above techniques, it is necessary to truncate the trajectories at some point in time. One solution is to partition the time dimension into "sessions" of a given length and keep a separate index for each session. However, this again introduces problems for updates and storage.

Obviously, these methods can be used to handle any query type for which spatial access methods are applicable [4,7,12,13,14].

2. Implicit Space-Time (Or Time Parameterized Space) Representation The objects' trajectories are represented as functions of time, which means, for instance, that only the speed and initial location vectors are stored in case of linear motion. The indexing is done with respect to the original spacial coordinates, but the index itself is made dependent on time.

Perhaps the most notable approaches in this category are the following:

Time Parameterized R-Trees (TPR Trees) of [14]: Here, the underlying structure is a usual R-tree, but one in which the minimum bounding rectangles are replaced by the so-called *conservative* bounding rectangles. Each such rectangle grows with time: at time t_{ref}, the rectangle is a minimum bounding rectangle for all moving objects enclosed inside it and continues to be a bounding rectangle for these objects (but not necessarily minimum) at any time $t \geq t_{\text{ref}}$. This is achieved by making the lower and upper bounds of the rectangle move, respectively, with the minimum and maximum speeds of the enclosed objects. Formally, assuming linear motion, a bounding rectangle enclosing a subset of moving objects $\mathcal{O}' \subseteq \mathcal{D}$ is determined by its lower and upper bounds, given respectively by $L(t) = \mathbf{a}_{\min}+\mathbf{v}_{\min}(t-t_{\text{ref}})$ and $U(t) = \mathbf{a}_{\max}+\mathbf{v}_{\max}(t-t_{\text{ref}})$, where the i-th components of $\mathbf{a}_{\min}$ and $\mathbf{v}_{\min}$ are given respectively by $\min\{o.y_i(t_{\text{ref}}) : o \in \mathcal{O}'\}$ and $\min\{o.v_i : o \in \mathcal{O}'\}$, for $i = 1,\ldots, d$, and $\mathbf{a}_{\max}$ and $\mathbf{v}_{\max}$ are defined similarly; see Fig. 3 for an illustration in the 1-dimensional case. As time passes, the parametrized rectangles may grow too much, thus affecting performance. To alleviate this problem, the rectangles are recomputed after a certain period of time, or when an object more enclosed in the rectangle is updated (this is called "*tightening*"); see Fig. 2a. Queries of types Q_1 and Q_2 can be answered by applying the standard R-tree query processing procedure. The difference is that, in this case, at each node of the tree, two time parameterized rectangles have to be checked for intersection. This can be done using a polyhedron-polyhedron intersection. However, a more direct procedure that takes advantage of the fact that these polyhedra are parameterized rectangles is given in [14]. Heuristics for packing the objects into



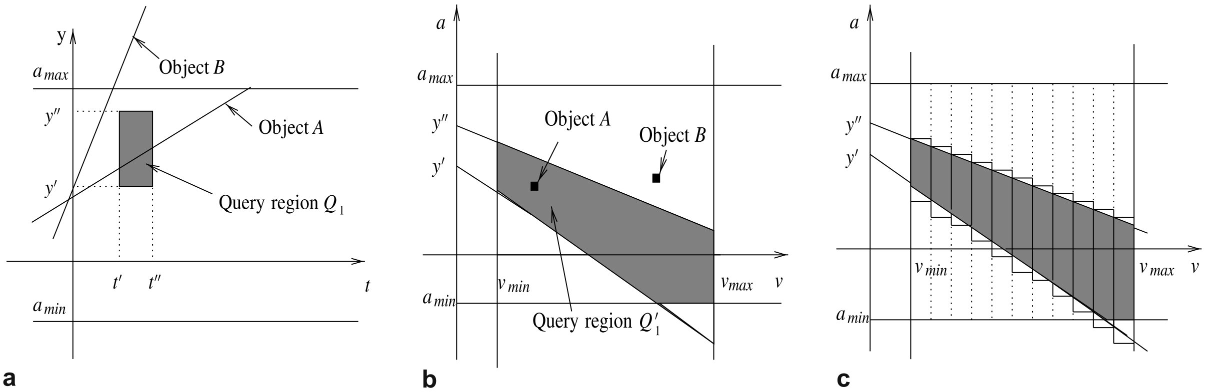
Indexing Schemes for Multi-dimensional Moving Objects, Figure 2 The TPR and B^x-trees: **a** The TPR-tree for the four objects at time t_1 and the tightened tree at t_2. **b** The B^x-tree: the small square represents the query region at the time the query is issued, while the large one represents the expanded query region (at t_{ref}); solid circles represent the locations of objects at t_{ref} and hollow circles represent the locations at the query time. The hollow circle at the bottom-right corner of the large square is a false hit

the nodes of a TPR-tree follow the corresponding ones for the regular R^*-tree [2], with the exception that the measure function (e. g., total volume of bounding rectangles, or overlap among them) is now a function of time whose *integration* over the lifetime of the index is to be minimized. In [11], an index called the *STAR-tree* was suggested to reduce the problem of frequent tightening by using some geometric ideas to compute *tighter* bounding rectangles than the conservative ones. This index is *self-adjusting* in the sense that no explicit tightening is needed (though at the cost of extra space requirement).

B^x-Tree and Its Variants Another approach used in [6] is to map the data into a 1-dimensional space and then use any 1-dimensional indexing structure, such as B^+-trees. In the *B^x-tree* approach, the time coordinate is partitioned into intervals, each of equal size to some upper bound Δ on the duration between two consecutive updates of any object's location. Each such interval is further partitioned into f phases for some integer $f > 0$. This way an object at most belongs to one of the phases at any given time. A B^+-tree is used to index the objects in each phase, where the keys are obtained by mapping the objects into a 1-dimensional space using a *space-filling curve*. More specifically, the d-dimensional data space is partitioned into a uniform grid whose cells are traversed using some space-filling curve; see Fig. 2b for an example using the *Hilbert* curve. This way each object is assigned a 1-dimensional key computed by concatenating its phase number and its space-filling-curve value, with the latter being computed using the location of the object at the start time t_{ref} of the phase. A query issued at time $t \geq t_{ref}$ is *expanded* conservatively using the minimum and maximum speeds of (all or a subset of) the objects to guarantee that all objects satisfying the query at the current time are reported. This may result in some *false positives* which can be removed in a post-processing step. Additionally, [6] suggested to reduce the number of false positives by maintaining a *histogram* of extremum speeds of objects in each grid cell. Depending on which extremum speeds are used for expanding the query range, different strategies can be obtained. For instance, in an iterative approach, one can start using the global extremum velocities to obtain an initial expanded region. Then the extremum velocities among the cells intersecting this region are used to get a refined expansion and so on. This technique has been used in [6] to answer queries of type Q_1, Q_2, and Q_4.

Achieving Logarithmic Query Time It should also be mentioned that if space is not an issue and is allowed to increase non-linearly with n, then logarithmic query time can be achieved; see, e. g., [7] and the *kinetic range trees* of [1].

3. Dual Space-Time Representation An alternative approach, assuming the objects are moving along linear trajectories, is to first map the objects into some *duality space* (see e. g., [5,7]) and then use some method to index the objects in this new space. The idea is to use a transformation under which the query answering problem reduces to a *simplex*, or more generally to a *polyhedral range searching* problem for which efficient indexing techniques are already known. One commonly used transformation is to map each d-dimensional object with velocity vector $\boldsymbol{v}$ and initial location $\boldsymbol{a}$ into a $2d$-dimensional point $(\boldsymbol{v}, \boldsymbol{a})$ (this is called the Hough-X transform in [5]). Given a query, say of type Q_1, the corresponding query region in the dual space can be obtained by eliminating the time variable t from the set of constraints defining Q_1 using, for example, the *Fourier-Motzkin elimination* method (see [3] for more details). For example, for the 1-dimensional version of query Q_1, under the existence of lower and upper bounds $v_{min} \leq |v| \leq v_{max}$ and $a_{min} \leq a \leq a_{max}$, the query region in the dual space becomes a (possibly) truncated trapezoid [7] (see Fig. 3). Hence, to retrieve all objects that pass between y' and y'' within the time interval $[t', t'']$, one needs to report all the points that lie inside the shaded trapezoid in Fig. 3b.



Indexing Schemes for Multi-dimensional Moving Objects, Figure 3 **a** Query Q_1 in native space. **b** Q_1 in Dual space. **c** Partitioning the dual query region for the MB-index

However, this does not generalize immediately to higher dimensions: if the above transform is used, query Q_1 transforms into a $2d$-dimensional region, bounded by a mix of linear and *quadratic* constraints [3]. This poses a problem since most currently known indexing techniques are designed to deal with polyhedral query regions, i. e., those that are described with linear constraints. One solution is to approximate the dual query region by the minimum bounding rectangle or by some other polyhedron, and add a post-processing step to filter out the objects not satisfying the query. Another solution, suggested in [3], is to use an alternative duality transformation which maps an object with velocity vector $\boldsymbol{v} = (v_1, \ldots, v_d)$ and initial location $\boldsymbol{a} = (a_1, \ldots, a_d)$ to a $2d$-dimensional point $(\boldsymbol{u}, \boldsymbol{w})$, where $u_i \overset{\text{def}}{=} 1/v_i$ and $w_i \overset{\text{def}}{=} a_i/v_i$, for $i = 1, \ldots, d$ (this is a variant of the so-called Hough-Y transform in [5]). For $v_i > 0$, this gives a one-to-one correspondence between (v_i, a_i) and (u_i, w_i). With such a transformation, query regions of type Q_1 and Q_2 become $2d$-dimensional polyhedra.

Indices With Proved Bounds on the Query Time The simplex range reporting problem has a rich background in the computational geometry literature. In particular, extending a known bound for main-memory range searching, [7] showed that simplex reporting in d-dimensions using only $O(n)$ disk blocks requires $\Omega(n^{1-1/d} + k)$ I/O's. In [8], an almost optimal main memory algorithm for simplex range searching is given using *partition trees*. Given a static set of N points in d-dimensions and an $O(N)$ memory space, this technique answers any simplex range query in $O(N^{1-1/d+\varepsilon} + K)$ time after an $O(N \log N)$ preprocessing time for any constant $\varepsilon > 0$. This has been adapted to the external memory model in [1,7], showing that a set of N points in two dimensions can be preprocessed in $O(N \log_B n)$ I/O's using $O(n)$ blocks as an *external partition tree*. Using such a structure, simplex range queries can be answered in $O(n^{1/2+\varepsilon} + k)$ I/O's while insertions and deletions can each be supported in amortized expected $O(\log_B^2 n)$ I/O's.

Briefly, a partition tree is a structure that represents a set of points in the Euclidean space; the root of the tree represents the whole set of points and the children of each node represent a *balanced simplicial partitioning* of the points represented by the parent node. A balanced simplicial partitioning of a point set S is a set of pairs $\{(S_1, \Delta_1), \ldots, (S_r, \Delta_r)\}$, where $S_1, \ldots, S_r$ form a partition of S, $|S_i| \le 2|S_j|$, for all $1 \le i, j \le r$, and each Δ_i is a simplex containing all the points in S_i. To guarantee efficient query performance, any straight line must cross at most $O(\sqrt{r})$ of the r simplices. Such partitioning can be constructed in $O(nr)$ I/O's [8]. In order to answer a simplex range query, one starts at the root. If a simplex lies entirely inside the query region, all the points in such a simplex are reported. If the simplex is not intersecting the query region, the simplex is discarded, otherwise the search is continued recursively with the children of the node representing this simplex. Combining the above duality transformation with partition trees, as suggested in [7], the 1-dimensional version of Q_1 (thereby mapped to a 2-dimensional dual space) can be answered in $O(n^{1/2+\varepsilon})$ I/O's using $O(n)$ disk blocks. For the d-dimensional version of Q_1, this technique would map the objects into a $2d$-dimensional dual space, and therefore answer such queries in $O(n^{1-1/2d+\varepsilon} + k)$ I/O's. In [1], another variant was suggested and yielded asymptotically better bounds for small dimensions by using *multilevel partition trees*. In particular, the asymptotic query time remains $O(n^{1/2+\varepsilon} + k)$ I/O's (assuming constant dimension), but the constant of proportionality increases exponentially with dimension. However, the hidden constant in the query search time of these techniques becomes large if a small ε is chosen.

Spatial Access Methods For more practical techniques, a spatial access method [4] may be used in the dual space. Although all such methods were originally designed to address orthogonal range queries (i. e., queries expressed as multi-dimensional hyper-rectangles), most of them can be easily modified to handle non-orthogonal queries. Note that the use of R*-trees in the dual space is equivalent to using a TPR-tree in the native space without tightening. Another approach, called *STRIPES*, which uses quad-trees in the dual space was suggested in [10].

Approximating the Query Region Another practical approach is based on the idea of approximating the query region in the dual space. This may result in some of the points (called *false hits*) not lying inside the original query region being processed before being excluded. For the 1-dimensional case of Q_1, [7] suggested the use of a rectangle for approximating the query region. In order to reduce the number of false hits, the coordinates of the objects in the original space are measured from c different "observation points", and c-indices are maintained corresponding to these points. It follows that, using this technique, such queries can be answered using $O(\log_B n + f + k)$ I/O's, where f is the number of I/O's resulting from false hits. Assuming uniformly distributed data, f can be bounded by n/c on the average. However, this approach uses $O(cn)$ disk blocks. Under non-uniform distributions of the parameters of the objects, c has to be large in order to reduce the number of false hits. In the *MB-index* of [3], the query region is approximated by partitioning the d-dimensional dual space into disjoint rectangular regions and using a B$^+$-tree to index the points in each region. In more details, this can be done by fixing one dimension, say the d-th dimension, and splitting the data points in each of the other dimensions $i = 1, \cdots, d-1$, using s $(d-1)$-dimensional hyperplanes perpendicular to the i-th dimension. Given a query region Q in the native space, the index is searched by first transforming the query region into one or more polyhedra in the dual space and then intersecting each such polyhedron with the s^{d-1} hyper-rectangular regions. The intersection defines s^{d-1} *conservative* lower and upper bounds on the d-th dimension, (see Fig. 3c), each of which is then used to search the corresponding B$^+$-tree. Finally, the resulting candidate answers are scanned to eliminate the false hits. By selecting s and the boundaries of each region appropriately, the number of false hits can be bounded. In particular, the 1-dimensional case of Q_1 can be answered on the average, assuming velocities and initial locations of objects are statistically independent (but not necessarily uniform), in $O(\sqrt{n \log_B n} + k)$ I/O's, with an amortized $O(\log_B n)$ I/O's per update, using $O(n)$ disk blocks. This can be generalized to queries of type Q_2 and Q_3, and also for higher dimensional versions of Q_1.

Key Applications

Transportation Industry

Recent progress in wireless communication technologies and geographical positioning systems (GPS) has made it possible, and relatively cheap, to connect moving vehicles to company databases. Being able to answer queries about these moving vehicles is one of the main motivations to build such databases, and efficient indexing techniques are important. For instance, for a database representing the location of taxi-cabs, a typical query may be: retrieve all free cabs that are currently within one mile of a given location, or retrieve the k nearest cabs (Q_1 or Q_4 queries). For a trucking company database, a typical query may be: retrieve all trucks that will be within one mile of a given truck (which needs assistance) within a certain time period (type Q_3 query).

Air-Traffic Control and Digital Battlefield

With the recently increased issues of safety and security, it is desirable to monitor all commercial aircrafts flying across the country at any given time. A database with this information can be queried, for instance, for all aircrafts that will approach a given region at a given point in time. The region could be either static or moving, or it could be one or more other aircrafts. This gives rise to queries of types Q_1, Q_2 or Q_3. Similar queries may also arise in digital battlefields in military applications.

Mobile Communication Systems

The use of mobile phones, palmtop computers, and smart cards has drastically increased recently. These usually work in a client-server model in which the servers are fixed base stations and the clients are the mobile devices. From the server point of view, it is useful for improving the quality of service to be able to predict either the number or the set of mobile devices that will approach the neighborhood of a given base station in a given time period.

Future Directions

Future work can address extensions of the existing methods to other types of queries; for instance, proximity queries of type Q_3 in higher dimensions. More techniques handling non-linear motion are of great interest. From the theoretical point of view, lower and upper bounds on the indexing problem, for $d \geq 2$, are still open. Another direction is to design practical indices with theoretically proved upper bounds, on the average, as was done preliminarily in [1,3].

Cross References

► Indexing, Hilbert R-tree, Spatial Indexing, Multimedia Indexing

Recommended Reading

1. Agarwal, P.K., Arge, L., Erickson, J.: Indexing moving points. J. Comput. Syst. Sci. **66**(1), 207–243 (2003)
2. Beckmann, N., Kriegel, H.P., Schneider, R., Seeger, B.: The R*-tree: an efficient and robust access method for points and rectangles. SIGMOD '90: Proceedings of the 1990 ACM SIGMOD international conference on Management of data (New York, NY, USA), ACM Press, pp. 322–331 (1990)
3. Elbassioni, K., Elmasry, A., Kamel, I.: An indexing method for answering queries on moving objects. Distrib. Parallel Databases **17**(3), 215–249 (2005)
4. Gaede, V., Günther, O.: Multidimensional access methods-, ACM Comput. Surv. **30**(2), 170–231 (1998)
5. Jagadish, H.V.: On indexing line segments, VLDB '90: Proceedings of the 16th International Conference on Very Large Data Bases San Francisco, CA, USA, pp. 614–625, Morgan Kaufmann Publishers Inc. (1990)
6. Jensen, C.S., Tiešytye, D., Tradišauskas, N.: Robust B+-tree-based indexing of moving objects, MDM '06: Proceedings of the 7th International Conference on Mobile Data Management (MDM'06) Washington, DC, USA, IEEE Computer Society, p. 12 (2006)
7. Kollios, G., Gunopulos, D., Tsotras, V.J.: On indexing mobile objects. PODS '99: Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. pp. 261–272 ACM Press, New York, NY, USA (1999)
8. Matoušek, J.: Efficient partition trees. Discrete Comput. Geom. **8**(3), 315–334 (1992)
9. Mokbel, M.F., Ghanem, T.M., Aref, W.G.: Spatio-temporal access methods. IEEE Data Eng. Bull. **26**(2), 40–49 (2003)
10. Patel, J.M., Chen, Y., Chakka, V.P.: STRIPES: an efficient index for predicted trajectories. SIGMOD '04: Proceedings of the 2004 ACM SIGMOD international conference on Management of data, pp. 635–646. ACM Press, New York, NY, USA (2004)
11. Procopiuc, C.M., Agarwal, P.K., Har-Peled, S.: STAR-tree: An efficient self-adjusting index for moving objects, ALENEX '02: Revised Papers from the 4th International Workshop on Algorithm Engineering and Experiments, pp. 178–193. Springer-Verlag, London, UK (2002)
12. Salzberg, B., Tsotras, V.J.: Comparison of access methods for time-evolving data, ACM Comput. Surv. **31**(2), 158–221 (1999)
13. Tayeb, J., Ulusoy, Ö, Wolfson, O.: A quadtree-based dynamic attribute indexing method. Comput. J. **41**(3), 185–200 (1998)
14. Šaltenis, S., Jensen, C.S., Leutenegger, S.T., Lopez, M.A.: Indexing the positions of continuously moving objects. SIGMOD '00: Proceedings of the 2000 ACM SIGMOD international conference on Management of data, pp. 331–342. ACM Press, New York, NY, USA (2000)

Indexing, Spatial

► Oracle Spatial, Geometries

Indexing Spatial Constraint Databases

PETER REVESZ
Department of Computer Science and Engineering, University of Nebraska Lincoln, Lincoln, NE, USA

Synonyms

Access structures for spatial constraint databases

Definition

Indexing structures are data structures used in computer science to store data. Indexing structures are closely associated with indexing methods or indexing algorithms that describe how to use the indexing structures correctly and efficiently to retrieve data, to insert data, or to modify or update data.

Main Text

Sometimes retrieval only means checking whether a given record is in the indexing structure or not. In this case only a "yes" or "no" answer needs to be returned. However, the retrieval of data is usually done on an associative basis, that is, a complex data with multiple components is to be retrieved based on one or more components that are given. For example, from an index structure of customer records, one may want to retrieve all those customer records which contain the street name "Apple Street." There are more sophisticated retrieval problems where instead of exact match between the records and the given information, a comparison or an approximate match is required. For example, retrieve the records of those customers who live within a mile from a specific store location, or retrieve the records of those customer(s) who live closest to the store location.

Sometimes the goal of retrieval is not simply to retrieve the records themselves but to retrieve some aggregate information or statistics of the relevant records. For example, one may be interested in the number of customers who live within a mile from a specific store location. This is called a *Count* query. Another example is finding the maximum number of airplanes that are within the airspace surrounding an airport at any given time. This is called a *Max-Count* query. When precise answers cannot be efficiently computed for Count or Max-Count aggregations, then it may be possible to develop algorithms that efficiently estimate their values.

When static spatial objects need to be indexed, then indexing spatial constraint databases can be done similarly to indexing spatial data or GIS data. However, for indexing

I

changing maps or moving objects, then the indexing structures are are more specialized. Cai and Revesz [2] extend the R-tree spatial data indexing structure to a parametric R-tree data structure that uses t as a temporal parameter to index moving objects. Revesz and Chen [3,5] describe count and max-count aggregation algorithms for moving objects. These have been extended with better estimation of Count and Max-Count in [1,4].

Cross References

- ► Constraint Databases, Spatial
- ► MLPQ Spatial Constraint Database System

Recommended Reading

1. Anderson, S.: Aggregation estimation for 2D moving points. 13th International Symposium on Temporal Representation and Reasoning, IEEE Press, Washington (2006)
2. Cai, M., Revesz, P.: Parametric R-Tree: An index structure for moving objects. 10th International Conference on Management of Data, pp. 57–64. McGraw-Hill, New Dehli (2000)
3. Chen, Y., Revesz, P.: Max-count aggregation estimation for moving points. 11th International Symposium on Temporal Representation and Reasoning,pp. 103–108. IEEE Press, Washington (2004)
4. Revesz, P.: Efficient rectangle indexing algorithms based on point dominance. 12th International Symposium on Temporal Representation and Reasoning, pp. 210–212. IEEE Press, Washington (2005)
5. Revesz, P., Chen, Y.: Efficient aggregation over moving objects. 10th International Symposium on Temporal Representation and Reasoning, pp. 118–127. IEEE Press, Washington (2003)

Indexing Spatio-temporal Archives

Marios Hadjieleftheriou[1], George Kollios[2], Vassilis J. Tsotras[3], Dimitrios Gunopulos[3]

[1] AT&T Inc., Florham Park, NJ, USA

[2] Computer Science Department, Boston University, Boston, MA, USA

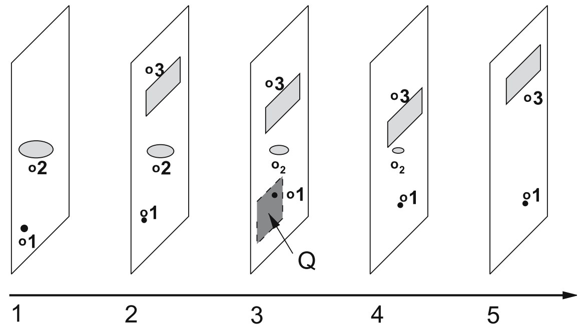
[3] Computer Science Department, University of California Riverside, Riverside, CA, USA

Synonyms

Spatio-temporal index structures; Moving objects; Lifetime; Evolution; Indexing, native-space; Indexing, parametric space; Index, R-tree; Index, MVR-tree

Definition

Consider a number of objects moving continuously on a 2-dimensional universe over some time interval. Given the complete archive of the spatio-temporal evolution of these objects, we would like to build appropriate index structures for answering range and nearest neighbor queries efficiently. For example: "Find all objects that appeared inside area S during time-interval $[t_1, t_2)$ " and "Find the object nearest to point P at time t".



Indexing Spatio-temporal Archives, Figure 1 A spatio-temporal evolution of moving objects

Consider the spatio-temporal evolution appearing in Fig. 1. The X and Y axes represent the plane while the T axis corresponds to the time dimension. At time 1 objects o_1 (a point) and o_2 (a region) appear. At time 2, object o_3 appears, while o_1 moves to a new position and o_2 shrinks. Object o_1 moves again at times 4 and 5; o_2 continues to shrink and disappears at time 5. Based on its behavior in the spatio-temporal evolution, each object is assigned a record with a *lifetime* interval $[t_i, t_j)$ according to when an object first appeared and disappeared. For example, the lifetime of o_2 is $L_2 = [1, 5)$. A spatio-temporal archive that stores the detailed evolution of the 2-dimensional universe consists of the complete update history of all the objects. A range query is also illustrated in the same figure: "Find all objects that appeared inside area Q at time 3"; only object o_1 satisfies this query. A variety of spatio-temporal and multi-dimensional index structures like the R-tree and its variants [11] can be used for indexing the spatio-temporal archive for the purpose of answering range and nearest neighbor queries. Nevertheless, given the special role that the temporal dimension plays in a spatio-temporal archive, it is possible to design methods and algorithms that can significantly improve upon existing approaches.

Historical Background

Due to the widespread use of sensor devices, mobile devices, video cameras, etc., large quantities of spatio-temporal data are produced everyday. Examples of applications that generate such data include intelligent transportation systems (monitoring cars moving on road networks), satellite and GIS analysis systems (evolution of forest boundaries, fires, weather phenomena), cellular network applications, video surveillance systems, and more.

Given the massive space requirements of spatio-temporal datasets it is crucial to develop methods that enable efficient spatio-temporal query processing. As a result, a number of new index methods for spatio-temporal data have been developed. They can be divided in two major categories: Approaches that optimize queries about the *future positions* of spatio-temporal objects (including continuous queries on the location of moving objects), and those that optimize *historical* queries, i. e., queries about past states of the spatio-temporal evolution. This article concentrates on historical queries.

Güting et al. [10] discussed the fundamental concepts of indexing spatio-temporal objects. Kollios et al. [15] presented methods for indexing the history of spatial objects that move with linear functions of time. The present article is an extension of this work for arbitrary movement functions. Porkaew et al. [22] proposed techniques for indexing moving points that follow trajectories that are combinations of piecewise functions. Two approaches were presented: The Native Space Indexing, where a 3-dimensional R-tree was used to index individual segments of the movement using one MBR per segment (what was referred to in this article as the piecewise approach), and the Parametric Space Indexing, which uses the coefficients of the movement functions of the trajectories to index the objects in a dual space (where only linear functions can be supported by this approach), augmented by the time-interval during which these movement parameters were valid, in order to be able to answer historical queries. A similar idea was used by Cai and Revesz [4]. The present work indexes the trajectories in the native space—being able to support arbitrary movement functions—and is clearly more robust than the piecewise approach in terms of selecting a query-efficiency vs. space tradeoff. Aggarwal and Agrawal [1] concentrated on nearest neighbor queries and presented a method for indexing trajectories with a special convex hull property in a way that guarantees query correctness in a parametric space. This approach is limited to specific classes of object trajectories and is targeted for nearest neighbor queries only. Pfoser et al. [21] introduced the TB-tree which is an indexing method for efficient execution of navigation and historical trajectory queries. TB-trees are optimized for trajectory preservation, targeting queries that need the complete trajectory information to be retrieved in order to be evaluated, in contrast to conventional range and nearest neighbor queries that need to acquire only partial information. Finally, Zhu et al. [27] proposed the Octagon-Prism tree which indexes trajectories by using octagon approximations. The Octagon-Prism tree is mostly related to the TB-tree.

A number of techniques for approximating 1-dimensional sequences have appeared in time-series research. Faloutsos et al. [9] presented a sliding window technique for grouping streaming values into sets of MBRs in order to approximate the time series and reduce the size of the representation. Keogh et al. [13] presented similar algorithms to the ones discussed here for segmenting time-series consisting of piecewise linear segments in an on-line fashion.

The problem of approximating piecewise linear 2-dimensional curves is of great importance in computer graphics and has received a lot of attention from the field of computational geometry as well, for at least the past thirty years. Kolesnikov [14] presents a concise analysis of all the algorithms that have been proposed in the past, including dynamic programming and greedy solutions like the ones discussed here. The pivotal work on this problem was introduced by Douglas and Peucker [6] and Pavlidis and Horovitz [20]. The work in [12] exploited these algorithms in 3-dimensional spaces and also introduced new algorithms for distributing a number of MBRs to a set of 3-dimensional curves.

Methods that can be used to index static spatio-temporal datasets include [5,15,18,23,24,25]. Most of these approaches are based either on the overlapping [3] or on the multi-version approach for transforming a spatial structure into a partially persistent one [2,7,17,26].

Scientific Fundamentals

Preliminaries

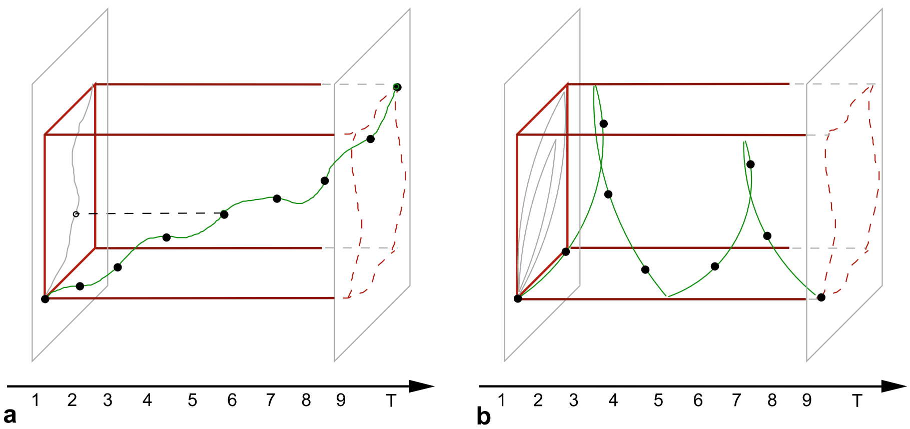
Indexing Alternatives for Spatio-temporal Archives The straightforward solution for this problem is to use any multi-dimensional index structure, like the R-tree and its variants [11]. An R-tree would approximate the whole spatio-temporal evolution of an object with one Minimum Bounding Region (MBR) that tightly encloses all the locations occupied by the object during its lifetime. While simple to deploy, simplistic MBR approximations introduce a lot of empty volume, since objects that have long lifetimes correspond to very large MBRs. This, in turn, introduces excessive overlapping between the index nodes of the tree and, therefore, leads to decreased query performance.

A better approach for indexing a spatio-temporal archive is to use a multi-version index, like the MVR-tree [8,17,26]. This index "logically" stores all the past states of the data evolution and allows updates only to the most recent state. The MVR-tree divides long-lived objects into smaller, better manageable intervals by introducing a number of object copies. A historical query is directed to the exact state acquired by the structure at the time that the query refers to; hence, the cost of answering the query is proportional only to the number of objects that the structure contained at that time. The MVR-tree is an improvement over the straightforward R-tree approach, especially for short time-interval queries, since the individual parts of the MVR-tree corresponding to different versions of the data that will be accessed for answering a historical query are more compact than an R-tree that indexes the whole evolution. However, there is still space for substantial improvement. Object clustering performed by the multi-version structure is affected by both the length of the lifetime of the objects and their spatial properties. By using rough MBR approximations with the multi-version structure we take advantage of the temporal dimension of the data only and not their spatial characteristics. Given that the spatio-temporal archive is available before building the index, it is possible to further improve index quality by creating finer object approximations that take into account the spatial dimensions as well.

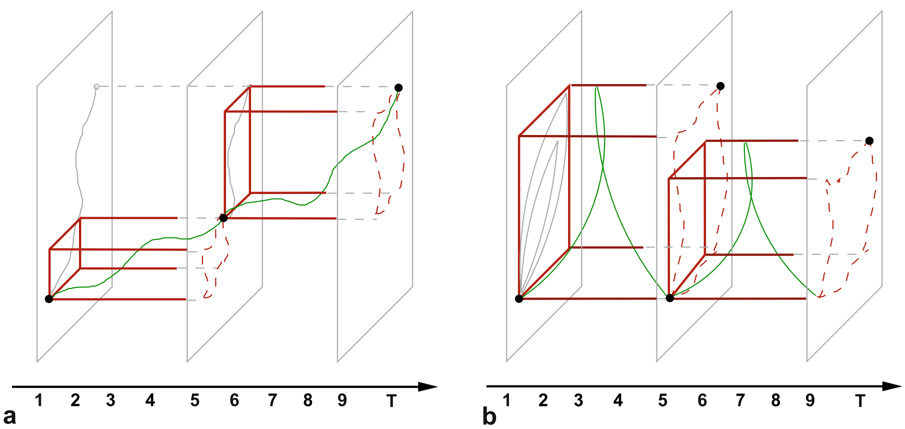
A number of algorithms can be applied on any spatio-temporal archive as a preprocessing step in order to improve index quality and, hence, query performance. These algorithms produce finer object approximations that limit empty volume and index node overlapping. Given N object trajectories the input to the algorithms are N MBRs, one per object trajectory. The output is K MBRs (typically $K = O(N)$) that approximate the original objects more accurately. Some of the N original MBRs may still be among the resulting K (for objects for which no better approximation was needed to be performed), while others will be split into sets of smaller, consecutive in time MBRs. The resulting MBRs will then be indexed with the aim of answering historical queries efficiently. Using these algorithms in combination with normal indexing techniques like the R-tree will not yield any improvements. Motivated by the query cost formula introduced by Pagel et al. [19] which states that the query performance of any MBR based index structure is directly proportional to the total volume, the total surface and the total number of indexed objects, it is argued that the proposed algorithms will benefit mostly the multi-version approach. This is due to the fact that introducing finer object approximations in the R-tree decreases the total volume of tree nodes but, at the same time, increases the total number of indexed objects, ripping off all benefits. On the other hand, in the MVR-tree the number of alive objects per time-instant remains constant independent of the number of MBRs used to approximate the objects, and this fact constitutes the biggest advantage of the multi-version approach.

Spatio-temporal Object Representation Object movements on a 2-dimensional plane can be modeled using various representations. Most applications represent a moving object as a collection of locations sampled every few seconds: The object movement is a set of tuples $\{\langle t_1, p_1\rangle, \dots, \langle t_n, p_n\rangle\}$. Another common assumption is that objects follow linear or piecewise linear trajectories. An object movement is then represented by a set of tuples $\{\langle [t_1, t_i), f_{x_1}(t), f_{y_1}(t)\rangle, \dots, \langle [t_j, t_n), f_{x_n}(t), f_{y_n}(t)\rangle\}$, where t_1 is the object insertion time, t_n is the object deletion time, $t_i, \dots, t_j$ are the intermediate time instants when the movement of the object changes characteristics and $f_{x_i}, f_{y_i} : \mathbb{R} \rightarrow \mathbb{R}$ are the corresponding linear functions. In general, the time-interval between two object updates is arbitrary, meaning that for any two consecutive time-instants t_i, t_j the corresponding time-interval $[t_i, t_j]$ can have an arbitrary length. In the degenerate case, the object can be represented by one movement function per time-instant of its lifetime. If objects follow complex non-linear movements this approach becomes inefficient as it requires a large number of linear segments in order to accurately represent the movement. A better approach is to describe movements by using combinations of more general functions. All these approaches can be easily extended for higher dimensionality.

Regardless of the preferred representation, an object can always be perceived as the collection of the locations it occupied in space for every time-instant of its lifetime. This *exact representation* is clearly the most accurate description of the object's movement that can be attained but, at the same time, it has excessive space requirements. Nevertheless, it is amenable to substantial compression. Given a space constraint or a desired approximation accuracy, one can derive various approximations of a spatio-temporal object. On the one extreme, the object can be approximated with a single MBR—a 3-dimensional box whose height corresponds to the object's lifetime interval and its base to the tightest 2-dimensional MBR that encloses all locations occupied by the object (in the rest, this representation is referred to as the *single MBR approximation*). This simplistic approximation introduces unnecessary empty volume. For example, in Fig. 2a point starts at time 1 from the lower left corner of the plane and follows an irregular movement. It is evident that approximating the point's movement with a single MBR introduces too much empty volume (but the space requirements of this approximation are only the two endpoints of the single MBR). A similar example is shown in Fig. 2b where a point moves in circles. On the other extreme, the exact representation is equivalent to keeping one point per time instant of the object's lifetime (for a total of nine points in Fig. 2a and b). This representation yields the maximum reduction in empty volume, but also the maximum number of required points, increasing space consumption substantially. Note that these figures depict a moving point for simplicity. In general, objects that have extents that vary over time would require one



Indexing Spatio-temporal Archives, Figure 2 Single MBR approximations. **a** An irregular movement, **b** a circular movement



Indexing Spatio-temporal Archives, Figure 3 Multiple MBR approximations. **a** An irregular movement, **b** a circular movement

MBR, instead of one point, per time instant of their lifetime.

Alternatively, spatio-temporal objects can be represented using multiple MBRs per object, which is a compromise in terms of approximation quality and space requirements. For example, in Fig. 3a and b two smaller MBRs are used to represent each trajectory. It is obvious that the additional MBRs increase the representation's space requirements. It is also apparent that there are many different ways to decompose a trajectory into consecutive MBRs, each one yielding different reduction in empty volume. Given a trajectory and a space constraint (imposed as a limit on the total number of MBRs), an interesting problem is to find the set of MBRs that produces the best approximation possible, given some optimality criterion (for instance, the minimization of the total volume of the representation). Notice that, it is not clear by simple inspection which are the best two MBRs for optimally reducing the total volume of the representation of the object depicted in Fig. 3b. Finding the optimal K MBRs for a given object trajectory is a difficult problem. Moreover, assume that we are given a set of trajectories and a space constraint (as a total number of MBRs). Another interesting problem is how to decide which objects need to be approximated more accurately than others (i. e., which objects need to be approximated using a larger number of MBRs) given that there is an upper-bound on the number of MBRs that can be used overall for the whole set.

The Multi-Version R-tree This section presents the MVR-tree, a multi-version indexing structure based on R-trees. Since this structure is an essential component of the proposed techniques, it is presented here in detail. Given a set of K 3-dimensional MBRs, corresponding to a total of $N < K$ object trajectories, we would like to index the MBRs using the MVR-tree. The MBRs are perceived by the tree as a set of K insertions and K deletions on 2-dimensional MBRs, since the temporal dimension has a special meaning for this structure. Essentially, an MBR with projection S on the X–Y plane and a lifetime interval equal to $[t1, t2)$ on the temporal dimension represents a 2-dimensional MBR S that is inserted at time t_1 and marked as logically deleted at time t_2.

Consider the sequence of $2 \cdot K$ updates ordered by time. MBRs are inserted/deleted from the index following this order. Assume that an ephemeral R-tree is used to index the MBR projections S that appeared at $t = 0$. At the next time-instant that an update occurs (a new MBR appears or an existing MBR disappears), this R-tree is updated by inserting/deleting the corresponding MBR. As time proceeds, the R-tree evolves. The MVR-tree conceptually stores the evolution of the above ephemeral R-tree over time. Instead of storing a separate R-tree per time-instant—which would result in excessive space overhead—the MVR-tree embeds all the states of the ephemeral R-tree evolution into a graph structure that has linear overhead to the number of updates in the evolution.

The MVR-tree is a directed acyclic graph of nodes. Moreover, it has multiple root nodes each of which is responsible for recording a consecutive part of the ephemeral R-tree evolution (each root splits the evolution into disjoint time-intervals). The root nodes can be accessed through a linear array whose entries contain a time-interval and a pointer to the tree that is responsible for that interval. Data records in the leaf nodes of an MVR-tree maintain the temporal evolution of the ephemeral R-tree data objects. Each data record is thus extended to include the lifetime of the object: *insertion-time* and *deletion-time*. Similarly, index records in the directory nodes of an MVR-tree maintain the evolution of the corresponding index records of the ephemeral R-tree and are also augmented with the same fields.

The MVR-tree is created incrementally following the update sequence of the evolution. Consider an update at time t. The MVR-tree is traversed to locate the target leaf node where the update must be applied. This step is carried out by taking into account the lifetime intervals of the index and leaf records encountered during the update. The search visits only the records whose lifetime fields contain t. After locating the target leaf node, an insertion adds the new data record with lifetime $[t, \infty)$, and a deletion updates the deletion-time of the corresponding data record from ∞ to t.

With the exception of root nodes, a node is called *alive* for all time instants that it contains at least $B \cdot P_v$ records that have not been marked as deleted, where $0 < P_v \leq 0.5$ and B is the node capacity. Otherwise, the node is considered *dead* and it cannot accommodate any more updates. This requirement enables clustering the objects that are alive at a given time instant in a small number of nodes, which in turn improves query I/O.

An update leads to a *structural change* if at least one new node is created. *Non-structural* are those updates which are handled within an existing node. An insertion triggers a structural change if the target leaf node already has B records. A deletion triggers a structural change if the target node ends up having less than $B \cdot P_v$ alive records. The former structural change is a *node overflow*, while the latter is a *weak version underflow* [2].

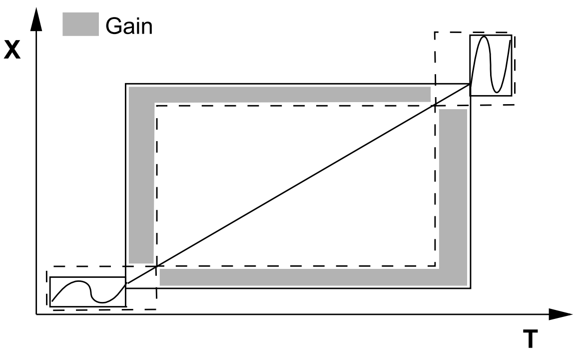
Node overflow and weak version underflow require special handling. Overflows cause a *split* on the target leaf node. Splitting a node A at time t is performed by creating a new node A' and copying all the alive records from A to A'. Now, node A can be considered dead after time t; any queries that refer to times later than t will be directed to node A' instead of A. To avoid having a structural change occurring too soon on node A', the number of alive entries that it contains when it is created should be in the range $[B \cdot P_{svu}, B \cdot P_{svo}]$, where P_{svu} and P_{svo} are predetermined constants. This allows a constant number of non-structural changes on A' before a new structural change occurs. If A' has more than $B \cdot P_{svo}$ alive records a *strong version overflow* occurs; the node will have to be *key-split* into two new nodes. A key-split does not take object lifetimes into account. Instead, it divides the entries according to their spatial characteristics (e. g., by using the R*-tree splitting algorithm which tries to minimize spatial overlap). On the other hand, if A' has less than $B \cdot P_{svu}$ alive records a *strong version underflow* occurs; the node will have to be merged with a sibling node before it can be incorporated into the structure (Kumar et al. [16] discuss various merging policies in detail).

Object Trajectory Approximation Algorithms

This section presents various algorithms for deciding how to approximate the objects contained in a spatio-temporal archive in order to reduce the overall empty volume, improve indexing quality and thus enhance query performance. Henceforth, the optimality of an approximation is considered only in terms of total volume reduction. The problem can be broken up into two parts:

1. Finding optimal object approximations: Given a spatio-temporal object and an upper limit on the number of MBRs that can be used to approximate it, find the set of consecutive MBRs that yield the representation with the minimum volume.
2. Reducing the overall volume of a dataset given a space constraint: Given a set of object trajectories and a space constraint (or, equivalently, a maximum number of MBRs) approximate each object with a number of MBRs such that the overall volume of the final set of MBRs is reduced.

Finding Optimal Object Approximations A simple method for approximating a spatio-temporal object using MBRs is to consider only the time instants when its movement/shape is updated (e. g., the functional boundaries) and partition the object along these points. Although, a few



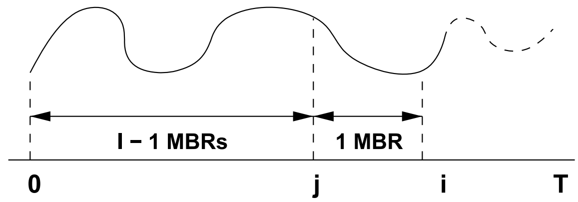
Indexing Spatio-temporal Archives, Figure 4 Approximating the object using the three dashed MBRs yields more reduction in empty volume than the piecewise approach

judiciously chosen MBRs are considerably more efficient in decreasing empty volume as shown in Fig. 4 with an example. Another way would be to let the MVR-tree split the object trajectories at the time instants when a leaf version split occurs. The resulting MBRs could be tightened directly after the split by looking at the raw trajectory data. Nevertheless, this technique would not yield the optimal per object approximations, neither can it be used to approximate some objects better than others. In addition, it has the added cost of reading the raw trajectory data multiple times during index creation. Therefore, more sophisticated approaches for finding optimal object approximations are essential.

Given a continuous time domain there are infinite number of points that can be considered as MBR boundaries. Although, practically, for most applications a minimum time granularity can be determined. Under this assumption, an appropriate granularity can be selected according to various object characteristics. The more agile an object is (i. e., the more complex the object movement is), the more detailed the time granularity that needs to be considered. On the other hand, objects that evolve very slowly can be accurately approximated using very few MBRs and thus a very coarse granularity in this case is sufficient. Of course, the more detailed the granularity, the more expensive the algorithms become. A good compromise between computational cost and approximation accuracy per object is application dependent. For ease of exposition in the rest it is assumed that a time granularity has already been decided for a given object. Two algorithms are presented that can be used to find the object approximation that minimizes the empty volume.

*An Optimal Algorithm (*DYNAMICSPLIT*)*

Given a spatio-temporal object O evolving during the interval $[t_0, t_n)$ that consists of n time instants (implied by the chosen time granularity) the goal is to find how to optimally partition the object using k MBRs, such that the final volume of the approximation is minimized. Let $V_l[t_i, t_j]$ be the volume of the representation corresponding to the part of the spatio-temporal object between time instants t_i and t_j after using l optimal MBRs. Then, the following holds:



Indexing Spatio-temporal Archives, Figure 5 Iteratively finding the best representation for interval $[t_0, t_i)$ using l MBRs

$$V_l[t_0, t_i] = \min_{0 \leq j < i} \{V_{l-1}[t_0, t_j] + V_1[t_j, t_i]\} .$$

The formula is derived as follows: Suppose that the optimal solutions for partitioning the sub-intervals $[t_0, t_1), [t_0, t_2), \ldots, [t_0, t_{i-1})$ of the object using $l-1$ MBRs are already known. The goal is to find the optimal solution for partitioning interval $[t_0, t_i)$, using l MBRs. The algorithm sets the $(l-1)$-th MBR boundary on all possible positions $j \in [0, 1, \ldots, i-1]$, dividing the object movement into two parts: The optimal representation for interval $[t_0, t_j)$ using $l-1$ MBRs, which is already known by hypothesis, and interval $[t_j, t_i)$ using only one MBR (Fig. 5). The best solution overall is selected, which is the optimal partition of $[t_0, t_i)$ using l MBRs. These steps are applied iteratively until the required amount of MBRs k for the whole lifetime of the object $[t_0, t_n)$ is reached.

Using the above formula a dynamic programming algorithm that computes the optimal positions for k MBRs is obtained. In addition, the total volume of this approximation is computed (value $V_k[t_0, t_n]$).

Theorem 1. *Finding the optimal approximation of an object using k MBRs (i. e., the one that minimizes the total volume) can be achieved in $O(n^2 k)$ time, where n is the number of discrete time instants in the object's lifetime.*

A total of nk values of the array $V_l[t_0, t_i]$ $(1 \leq l \leq k, 0 \leq i \leq n)$ have to be computed. Each value in the array is the minimum of at most n values, computed using the above formula. The volume $V_1[j, i]$ of the object between positions j and i can be precomputed for every run i and all values of j using $O(n)$ space and $O(n)$ time and thus does not affect the time complexity of the algorithm. □

*An Approximate Algorithm (*GREEDYSPLIT*)*

The DYNAMICSPLIT algorithm is quadratic to the number of discrete time instants in the lifetime of the object.

For objects that live for long time-intervals and for very detailed time granularities the above algorithm is not very efficient. A faster algorithm is based on a greedy strategy. The idea is to start with n consecutive MBRs (the exact representation with the given time granularity) and merge the MBRs in a greedy fashion (Algorithm 1). The running time of the algorithm is $O(n \lg n)$, due to the logarithmic overhead for updating the priority queue at step 2 of the algorithm. This algorithm gives, in general, sub-optimal solutions.

Algorithm 1 GREEDYSPLIT
A spatio-temporal object O as a sequence of n consecutive MBRs, one for each time-instant of the object's lifetime.

InputOutput: A set of k MBRs that represent O's movement.

1: For $0 \leq i < n$ compute the volume of the resulting MBR after merging O_i with O_{i+1}. Store the results in a priority queue.
2: Repeat $n - k$ times: Use the priority queue to merge the pair of consecutive MBRs that give the smallest increase in volume. Update the priority queue with the new (merged) MBR.

Reducing the Overall Volume of a Dataset Given a Space Constraint It is apparent that given a set of objects, in order to represent all of them accurately, some objects may require only few MBRs while others might need a much larger number. Thus, it makes sense to use varying numbers of MBRs per object, according to individual object evolution characteristics. This section discusses methods for approximating a set of N spatio-temporal objects using a given total number of MBRs K such that the total volume of the final object approximations is minimized.[1] In the following, we refer to this procedure as the *MBR assignment process*, implying that, given a set of K "non-materialized" MBRs, each MBR is assigned to a specific object iteratively, such that the volume of the object after being approximated with the extra MBR is minimized (assuming that all objects are initially assigned only one MBR).

*An Optimal Algorithm (*DYNAMICASSIGN*)*
Assuming that the objects are ordered from 1 to N, let $MTV_l[i]$ be the Minimum Total Volume consumed by the first i objects with l optimally assigned MBRs and $V_k[i]$ be the total volume for approximating the i-th object using k MBRs. The following observation holds:

$$MTV_l[i] = \min_{0 \leq k \leq l} \{MTV_{l-k}[i-1] + V_k[i]\} .$$

[1] Where K is implied by some space constraint, e. g., the available disk space.

Intuitively, the formula states that if it is known how to optimally assign up to $l - 1$ MBRs to $i - 1$ objects, it can be decided how to assign up to l MBRs to i objects by considering all possible combinations of assigning one extra MBR between the $i - 1$ objects and the new object. The idea is similar to the one explained for the DYNAMICSPLIT algorithm. A dynamic programming algorithm can be implemented with running time complexity $O(NK^2)$. To compute the optimal solution the algorithm needs to know the optimal approximations per object, which can be computed using the DYNAMICSPLIT algorithm. Hence, the following theorem holds:

Theorem 2. *Optimally assigning K MBRs among N objects takes $O(NK^2)$ time.*
A total of NK values for array $MTV_l[i]$ ($0 \leq l \leq K, 1 \leq i \leq N$) need to be computed, where each value is the minimum of at most $K + 1$ values for $0 \leq k \leq K$, in each iteration.□

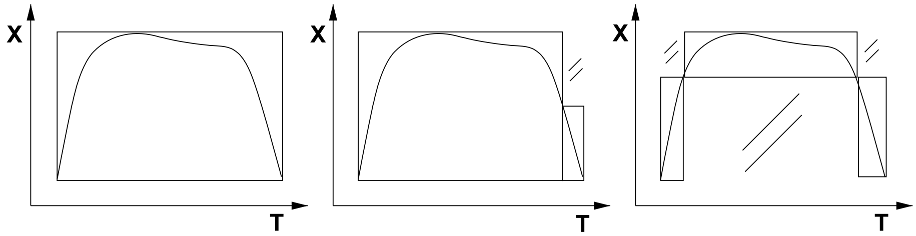
*A Greedy Algorithm (*GREEDYASSIGN*)*
The DYNAMICASSIGN algorithm is quadratic to the number of MBRs, which makes it impractical for large K. For a faster, approximate solution a greedy strategy can be applied: Given the MBR assignments so far, find the object that if approximated using one extra MBR will yield the maximum possible global volume reduction. Then, assign the MBR to that object and continue iteratively until all MBRs have been assigned. The algorithm is shown in Algorithm 2. The complexity of step 2 of the algorithm is $O(K \lg N)$ (the cost of inserting an object K times in the priority queue) thus the complexity of the algorithm itself is $O(K \lg N)$ (since it is expected that $N < K$).

Algorithm 2 GREEDYASSIGN
A set of N spatio-temporal objects and a number K.

Proof. **InputOutput:** A near optimal minimum volume required to approximate all objects with K MBRs.

1: Assume that each object is represented initially using a single MBR. Find what the volume reduction $VR_{i,2}$ per object i would be when using 2 MBRs to approximate it. Store in a max priority queue according to $VR_{i,2}$.
2: For K iterations: Remove the top element i of the queue, with volume reduction $VR_{i,k}$. Assign the extra k-th MBR to i. Calculate the reduction$VR_{i,k+1}$ if one more MBR is used for i and reinsertit in the queue.

*An Improved Greedy Algorithm (*LAGREEDYASSIGN*)*
The result of the GREEDYASSIGN algorithm will not be optimal in the general case. One reason is the following: Consider an object that yields a small empty volume reduction when approximated using only two MBRs, while most of its empty volume is removed when three MBRs are used



Indexing Spatio-temporal Archives, Figure 6 Two MBRs yield almost no reduction in empty space, while three MBRs reduce it substantially

(an 1-dimensional example of such an object is shown in Fig. 6). Using the GREEDYASSIGN algorithm it is probable that this object will not be given the chance to be assigned any MBRs at all, because its first MBR allocation produces poor results and other objects will be selected instead. However, if the algorithm is allowed to consider more than one assigned MBRs per object per step, the probability of assigning more MBRs to this object increases. This observation gives the intuition on how the greedy strategy can be improved. During each iteration, instead of choosing the object that yields the largest volume reduction by assigning only one more MBR, the algorithm can *look ahead* and find the object that results in even bigger reduction if two, three, or more MBRs were assigned all at once.

For example, the look-ahead-2 algorithm works as follows (Algorithm 3): First, all MBRs are assigned using the GREEDYASSIGN algorithm as before. Then, one new priority queue PQ_1 is created which sorts the objects by the volume reduction offered by their last assigned MBR (if an object has been assigned k MBRs, the object is sorted according to the volume reduction yielded by the k-th MBR). The top of the queue is the *minimum reduction*. A second priority queue PQ_2 is also needed which sorts each object by the volume that would be reduced if it was assigned two more MBRs (if an object has been assigned k MBRs, the object is sorted according to the volume reduction produced by $k + 2$ MBRs). The top of the queue is the *maximum reduction*. If the volume reduction of the top element of PQ_2 is bigger than the sum of the reduction of the two top elements of PQ_1 combined, the splits are reassigned accordingly and the queues are updated. The same procedure continues until there are no more redistributions of MBRs. In essence, the algorithm tries to find two objects for which the combined reduction from their last assigned MBRs is less than the reduction obtained if a different, third object, is assigned two extra MBRs. The algorithm has the same worst case complexity as the greedy approach. Experimental results show that it achieves much better results for the small time penalty it entails.

Algorithm 3 LAGREEDYASSIGN

A set of spatio-temporal objects with cardinality N and a number K.

Proof. **InputOutput:** A set of spatio-temporal objects with cardinality N and a number K.

1: Allocate MBRs by calling the GREEDYASSIGN algorithm. PQ_1 is a min priority queue that sorts objects according to the reduction given by their last assigned MBR. PQ_2 is a max priority queue that sorts objects according to the reduction given if two extra MBRs are used per object.

2: Remove the top two elements from PQ_1, let O_1, O_2. Remove the top element from PQ_2, let O_3. Ensure that $O_1 \neq O_2 \neq O_3$, otherwise remove more objects from PQ_1. If the volume reduction for O_3 is larger than the combined reduction for O_1 and O_2, redistribute the MBRs and update the priority queues.

3: Repeat last step until there are no more redistributions of MBRs.

Key Applications

All applications that generate spatio-temporal data would benefit significantly from using advanced historical index structures. A few examples are intelligent transportation systems (monitoring cars moving on road networks), satellite and GIS analysis systems (evolution of forest boundaries, fires, weather phenomena), cellular network applications, and video surveillance systems.

Future Directions

Future work could concentrate on investigating the online version of the problem. Given a stream of object updates, we would like to maintain a historical spatio-temporal structure efficiently given that the data is made available incrementally and a pre-processing step is not feasible. In particular, it would be interesting to explore how dynamically evolving object update distributions can be detected in order to develop appropriate buffering policies that can be used for pre-processing the data within the buffer, before being inserted into the historical index structure.

Cross References

► Indexing of Moving Objects, B^x-Tree
► Indexing the Positions of Continuously Moving Objects
► Indexing Schemes for Multi-dimensional Moving Objects

- ► Mobile Object Indexing
- ► Mobile Objects Databases
- ► Movement Patterns in Spatio-temporal Data
- ► Patterns in Spatio-temporal Data
- ► R*-tree
- ► R-Trees – A Dynamic Index Structure for Spatial Searching

Recommended Reading

1. Aggarwal, C.C., Agrawal, D.: On nearest neighbor indexing of nonlinear trajectories. In: PODS, pp. 252–259 (2003)
2. Becker, B., Gschwind, S., Ohler, T., Seeger, B., Widmayer, P.: An asymptotically optimal multiversion B-Tree. VLDB J. **5**(4), 264–275 (1996)
3. Burton, F., Kollias, J., Kollias, V., Matsakis, D.: Implementation of overlapping B-trees for time and space efficient representation of collection of similar files. Comput. J., **33**(3), 279–280 (1990)
4. Cai, M., Revesz, P.: Parametric R-tree: An index structure for moving objects. In: COMAD, pp. 57–64 (2000)
5. Chakka, V.P., Everspaugh, A., Patel, J.M.: Indexing large trajectory data sets with SETI. In: CIDR, pp. 164–175 (2003)
6. Douglas, D.H., Peucker, T.K.: Algorithms for the reduction of the number of points required to represent a digitised line or its caricature. Can. Cartogr. **10**(2), 112–122 (1973)
7. Driscoll, J., Sarnak, N., Sleator, D., Tarjan, R.E.: Making data structures persistent. In: STOC (1986)
8. Driscoll, J.R., Sarnak, N., Sleator, D.D., Tarjan, R.E.: Making data structures persistent. J. Comput. Syst. Sci. **38**(1), 86–124 (1989)
9. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time-series databases. In: SIGMOD, pp. 419–429 (1994)
10. Güting, R.H., Bhlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N.A., Schneider, M., Vazirgiannis, M.: A foundation for representing and querying moving objects. TODS **25**(1), 1–42 (2000)
11. Guttman, A.: R-trees: A dynamic index structure for spatial searching. In: SIGMOD, pp. 47–57 (1984)
12. Hadjieleftheriou, M., Kollios, G., Tsotras, V.J., Gunopulos, D.: Indexing spatiotemporal archives. VLDB J. **15**(2), 143–164 (2006)
13. Keogh, E.J., Chu, S., Hart, D., Pazzani, M.J.: An online algorithm for segmenting time series. In: ICDM, pp. 289–296 (2001)
14. Kolesnikov, A.: Efficient algorithms for vectorization and polygonal approximation. PhD thesis, University of Joensuu, Finland, (2003)
15. Kollios, G., Tsotras, V.J., Gunopulos, D., Delis, A., Hadjieleftheriou, M.: Indexing animated objects using spatiotemporal access methods. TKDE **13**(5), 758–777 (2001)
16. Kumar, A., Tsotras, V.J., Faloutsos, C.: Designing access methods for bitemporal databases. TKDE **10**(1), 1–20 (1998)
17. Lomet, D., Salzberg, B.: Access methods for multiversion data. In: SIGMOD, pp. 315–324 (1989)
18. Nascimento M., Silva, J.: Towards historical R-trees. In: SAC, pp. 235–240 (1998)
19. Pagel, B.-U., Six, H.-W., Toben, H., Widmayer, P.: Towards an analysis of range query performance in spatial data structures. In: PODS, pp. 214–221 (1993)
20. Pavlidis, T., Horovitz, S.L.: Segmentation of plane curves. IEEE Trans. Comput. **23**(8), 860–870 (1974)
21. Pfoser, D., Jensen, C.S., Theodoridis, Y.: Novel approaches in query processing for moving object trajectories. In: VLDB, pp. 395–406 (2000)
22. Porkaew, K., Lazaridis, I., Mehrotra, S.: Querying mobile objects in spatio-temporal databases. In: SSTD, pp. 59–78 (2001)
23. Tao, Y., Papadias, D.: MV3R-Tree: A spatio-temporal access method for timestamp and interval queries. In: VLDB, pp. 431–440 (2001)
24. Theodoridis, Y., Sellis, T., Papadopoulos, A., Manolopoulos, Y.: Specifications for efficient indexing in spatiotemporal databases. In: SSDBM, pp. 123–132 (1998)
25. Tzouramanis, T., Vassilakopoulos, M., Manolopoulos, Y.: Overlapping linear quadtrees and spatio-temporal query processing. Comput. J. **43**(3), 325–343 (2000)
26. Varman, P.J., Verma, R.M.: An efficient multiversion access structure. TKDE **9**(3), 391–409 (1997)
27. Zhu, H., Su, J., Ibarra, O.H.: Trajectory queries and octagons in moving object databases. In: CIKM, pp. 413–421 (2002)

Indexing the Positions of Continuously Moving Objects

SIMONAS ŠALTENIS
Department of Computer Science, Aalborg University, Aalborg, Denmark

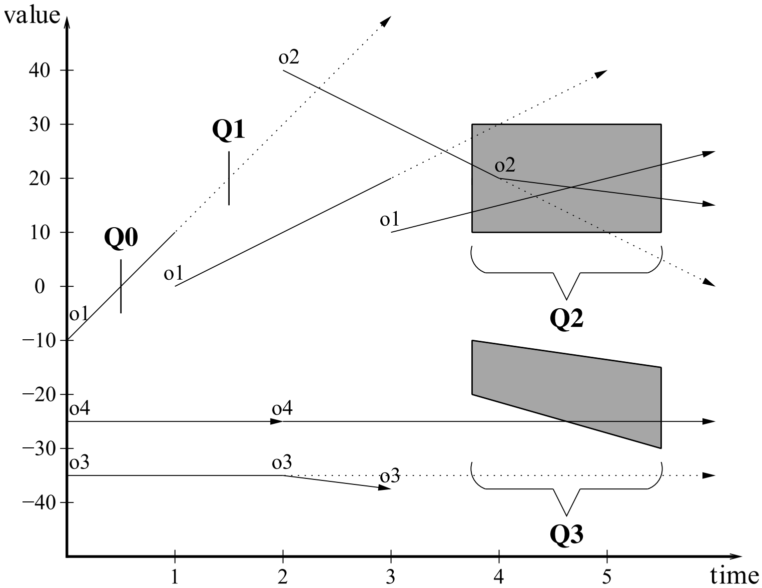
Synonyms

Indexing moving objects; Spatio-temporal indexing

Definition

Wireless communications and positioning technologies enable tracking of the changing positions of objects capable of continuous movement. Continuous movement poses new challenges to database technology. In conventional databases, data is assumed to remain constant unless it is explicitly modified. Capturing continuous movement under this assumption would entail either performing very frequent updates or recording outdated, inaccurate data, neither of which are attractive alternatives. Instead, rather than storing simple positions, functions of time that express the objects' changing positions are stored [21]. More specifically, linear functions are assumed. This entry describes indexing of the current and anticipated future positions of such objects with the focus on the TPR-tree [16].

Modeling the positions of moving objects as functions of time enables querying not only the current positions of objects but also tentative future predictions of these positions. The index should support queries that retrieve all points with positions within specified regions. It is possible to distinguish between three kinds of queries based on the regions they specify: *timeslice queries*, *window queries*, and *moving queries*. Figure 1 shows a set of trajectories



Indexing the Positions of Continuously Moving Objects, Figure 1 Query examples for one-dimensional data [16]

of one-dimensional moving objects and examples of the three types of queries. Here, queries *Q0* and *Q1* are time-slice queries, *Q2* is a window query, and *Q3* is a moving query.

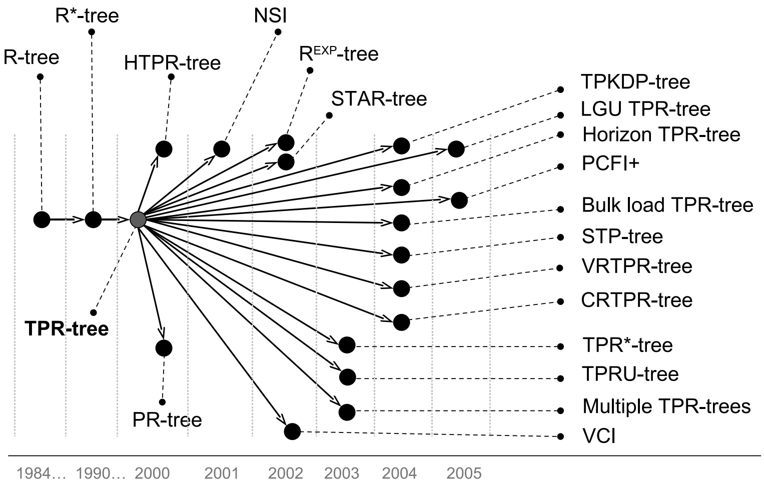
Similarly, with an additional input of a time point or a time interval, other types of spatial queries such as nearest neighbor queries or reverse nearest neighbor queries should be supported [4]. The index should also support insertions and deletions. An update is handled as a deletion followed by an insertion.

Historical Background

The concept of moving object databases and modeling of continuous movement as linear functions is rather recent. It was first introduced by Wolfson et al. in 1998 [21]. The first proposal for indexing such data was introduced by Tayeb et al. [19], who proposed using PMR-quadtrees [12] for indexing the future linear trajectories of one-dimensional moving point objects as line segments in (x, t)-space. Kollios et al. [8] also focuses mainly on one-dimensional data and employs the so-called *duality* data transformation where a line $x = x(t_{\text{ref}}) + v(t - t_{\text{ref}})$ is transformed to the point $((x(t_{\text{ref}}), v))$, enabling the use of regular spatial indices.

Later research focuses on two-dimensional and, in some cases, three-dimensional data. In general, the proposed indexing methods can be classified according to the space that they index, i. e., what view is taken on the indexed data. Assume the objects move in a d-dimensional space ($d = 1, 2, 3$). The first approach is to index future trajectories as lines in a $(d + 1)$-dimensional space. This approach is taken in the above-mentioned method by Tayeb et al. [19], but the method is difficult to extend to higher dimensions and the index has to be periodically rebuilt. The second approach is to map the trajectories to points in a higher-dimensional space which are then indexed. Queries must subsequently also be transformed to counter the data transformation. The above-mentioned duality transformation maps the d-dimensional linearly moving points into $2d$-dimensional static points [8]. In STRIPES [13], PR quadtrees [18] are used to index these $2d$-dimensional points. Yiu et al. instead proposed to use space filling curves to further transform $2d$-dimensional points into one-dimensional points that are then indexed by the B^+-tree. Finally, Agarwal et al. [1] combined the duality transformation with kinetic data structures [2]. The main idea of kinetic data structures is to schedule future events that update a data structure so that necessary invariants hold.

The third approach, sometimes referred to as indexing in the *primal* space, is to index data in its native, d-dimensional space, which is possible by parameterizing the index structure using velocity vectors and thus enabling the index to be "viewed" as of any future time. This absence of transformations yields quite intuitive indexing techniques. The Time Parameterized R-tree (TPR-tree) [16] is the main example of this approach. First proposed by Šaltenis et al. in 2000, the TPR-tree gave rise to a number of other access methods (see Fig. 2). For example, the R^{EXP}-tree extends the TPR-tree to index data with expiration times so that objects that do not update their positions are automatically removed from the index. The TPR*-tree [20] adds a number of heuristics to improve the query performance of the TPR-tree and to optimize it for a slightly different workload of queries than the one considered by the authors of the TPR-tree. The STAR-tree [15] modifies the TPR-tree



Indexing the Positions of Continuously Moving Objects, Figure 2 TPR-tree origins and the follow-up research

by introducing more complex time-parameterized bounding rectangles and making the index self-adjustable. Finally, the Velocity Constrained Indexing (VCI) [14] uses the regular R-tree [5] with an additional field of $v_{\max}$ added to each node. The $v_{\max}$ is used to expand bounding rectangles of the R-tree when future queries are asked.

The fourth approach is to index the objects' positions at some specific label timestamps and to extend a query according to the maximum speed of objects. B^x-tree [6,7] uses a combination of space-filling curves and B^+-trees to index the static positions of objects at label timestamps.

Scientific Fundamentals

Data and Queries

For an object moving in a d-dimensional space, the object's position at some time t is given by $\bar{x}(t) = (x_1(t), x_2(t), \ldots, x_d(t))$, where it is assumed that the times t are not before the current time. This position is modeled as a linear function of time which is specified by two parameters. The first is a position for the object at some specified time t_{ref}, $\bar{x}(t_{\text{ref}})$, which is called the reference position. The second parameter is a velocity vector for the object, $\bar{v} = (v_1, v_2, \ldots, v_d)$. Thus, $\bar{x}(t) = \bar{x}(t_{\text{ref}}) + \bar{v}(t - t_{\text{ref}})$.

Then, as shown in Fig. 1, *Timeslice query*, $Q = (R, t)$, retrieves points that will be inside the d-dimensional hyper-rectangle R at time t. *Window query*, $Q = (R, t^{\vdash}, t^{\dashv})$, retrieves points that will be inside the hyper-rectangle R sometime during time-interval $[t^{\vdash}, t^{\dashv}]$. *Moving query*, $Q = (R_1, R_2, t^{\vdash}, t^{\dashv})$, retrieves points with trajectories in $(\bar{x}, t)$-space crossing the $(d + 1)$-dimensional trapezoid obtained by connecting R_1 at time $t^{\vdash}$ to R_2 at time $t^{\dashv}$.

The Structure of the TPR-Tree

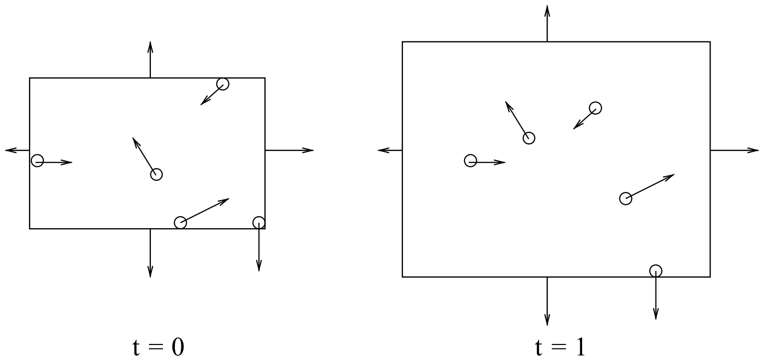
The TPR-tree indexes continuously moving points in one, two, or three dimensions. It employs the basic structure of the R-tree [5], which stores data in the leaves of a balanced index tree and each non-leaf index entry contains a minimum bounding rectangle (MBR) of all the data in the subtree pointed to by the entry. In contrast to the R-tree, both the indexed points and the bounding rectangles are augmented with velocity vectors in the TPR-tree. This way, bounding rectangles are time parameterized—they can be computed for different time points. Velocities are associated with the edges of bounding rectangles so that the enclosed moving objects, be they points or other rectangles, remain inside the bounding rectangles at all times in the future. More specifically, if a number of points p_i are bounded at time t, the spatial and velocity extents of a bounding rectangle along the x axis are computed as follows:

$$x^{\vdash}(t) = \min_i\{p_i.x(t)\}\,; \quad x^{\dashv}(t) = \max_i\{p_i.x(t)\}\,;$$
$$v_x^{\vdash} = \min_i\{p_i.v_x\}\,; \quad v_x^{\dashv} = \max_i\{p_i.v_x\}\,.$$

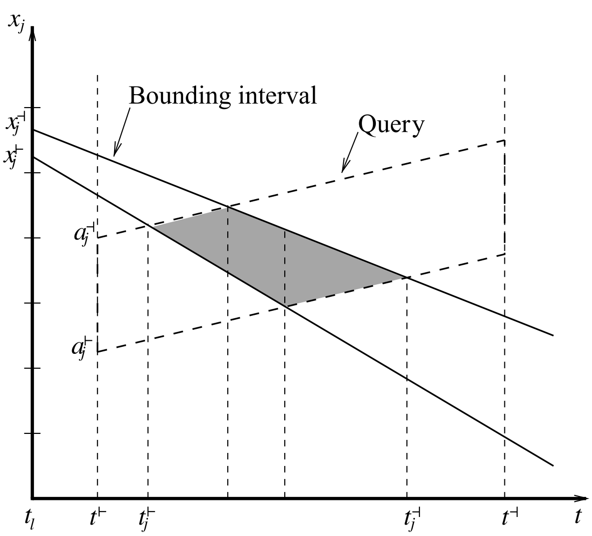
Figure 3 shows an example of the evolution of a bounding rectangle in the TPR-tree computed at $t = 0$. Note that, in contrast to R-trees, bounding rectangles in the TPR-tree are not minimum at all times. In most cases, they are minimum only at the time when they are computed. In a process called *tightening*, whenever an index node is modified during insertions or deletions, the node's time-parameterized bounding rectangle is recomputed, making it minimum at that time.

Querying the TPR-Tree

The TPR-tree can be interpreted as an R-tree for any specific time, t_q. This suggests that the algorithms that are based on the R-tree are easily "portable" to the TPR-tree. For example, answering a timeslice query proceeds as for the regular R-tree, the only difference being that all bound-



Indexing the Positions of Continuously Moving Objects, Figure 3 Example time-parameterized bounding rectangle [4]



Indexing the Positions of Continuously Moving Objects, Figure 4 Intersection of a bounding interval and a query [16]

ing rectangles are computed for the time t_q specified in the query before the intersection is checked.

To answer window queries and moving queries, the algorithm has to check if, in $(\bar{x}, t)$-space, the trapezoid of a query intersects with the trapezoid formed by the part of the trajectory of a bounding rectangle that is between the start and end times of the query. This can be checked using a simple algorithm [16]. Figure 4 demonstrates the intersection between a one-dimensional time parameterized bounding rectangle (interval) and a moving query.

Updating the TPR-Tree

An update of the moving object's position is modeled as a deletion of the old position followed by an insertion of a new position. The TPR-tree's update algorithms are based on the update algorithms of the R*-tree [3], which is an R-tree with improved update algorithms. The update algorithms of the TPR-tree differ from the corresponding algorithms of the R*-tree only in the heuristics that are used. The R*-tree uses heuristics that minimize certain functions, such as the area of a bounding rectangle, the intersection of two bounding rectangles, the margin of a bounding rectangle, and the distance between the centers of two bounding rectangles. The TPR-tree employs time-parameterized bounding rectangles which in turn means that the above-mentioned functions are time dependent, and their evolution in time should be considered. Specifically, given an objective function $A(t)$, the following integral should be minimized:

$$\int_{t_c}^{t_c+H} A(t)\,\mathrm{d}t\,,$$

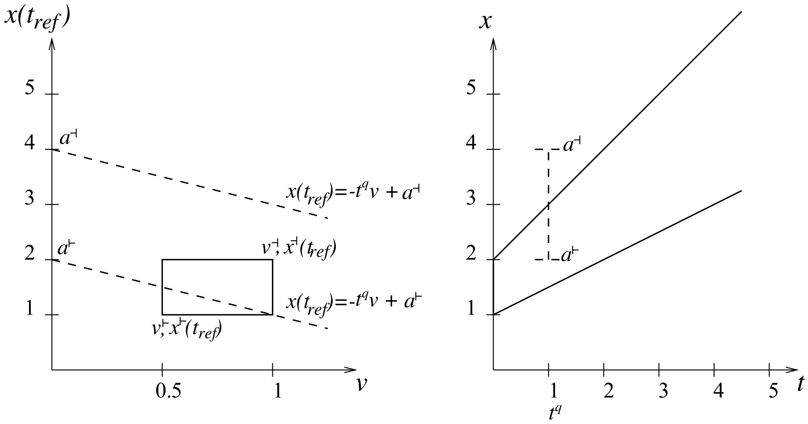
where t_c is the time when the heuristics is being applied and H is the so-called *horizon* parameter that determines how far into the future the queries will "see" the effects of the application of this heuristics. If $A(t)$ is area, the integral computes the area (volume) of the trapezoid that represents part of the trajectory of a bounding rectangle in $(\bar{x}, t)$-space (see Fig. 4).

Duality-Transformation Approach

The general approach of indexing moving points in their native space using a time-parameterized index structure such as the TPR-tree is very closely related to the duality transformation approach [8,13,23].

Considering one-dimensional data, the duality transformation transforms the linear trajectory of a moving point $x = x(t_{\mathrm{ref}}) + v(t - t_{\mathrm{ref}})$ in (x, t)-space into a point $(x(t_{\mathrm{ref}}), v)$, where t_{ref} is a chosen reference time. Then, queries are also transformed.

Bounding points $(x(t_{\mathrm{ref}}), v)$ in the dual space with a minimum bounding rectangle is equivalent to bounding them (as moving points) with a time-parameterized bounding interval computed at t_{ref}. Figure 5 shows the same bounding rectangle and query in $(x(t_{\mathrm{ref}}), v)$-space and in (x, t)-space.



Indexing the Positions of Continuously Moving Objects, Figure 5 Timeslice query (dashed) and bounding interval (solid) in dual $(x(t_{\mathrm{ref}}), v)$-space and (x, t)-space

In spite of such equivalence of bounding rectangles in both approaches, the algorithms used in specific indexes might be quite different. A duality-transformation index may not even explicitly use minimum bounding rectangles [23]. Furthermore, while the heuristics of time-parameterized indexes consider the objects' positions at the time the heuristics are applied, the algorithms of duality-transformation approaches always use a pre-chosen constant t_{ref}. For this reason, duality-transformation approaches usually use two indexes, such that newer updates are put into the second index and, when the first index becomes empty, the indexes change roles [8].

Note that a sufficiently high update rate is crucial for both the time-parameterized indexes, such as the TPR-tree, and the indexes using the duality transformation. If the rate of updates is too low, the indexes suffer the degradation of query performance. The reasons for this are most obvious in the TPR-tree, where the degradation is caused by the expansion of time-parameterized bounding rectangles, resulting in more queries intersecting with a given bounding rectangle.

Key Applications

Online, Position-Aware People, Vehicles, and Other Objects

The rapid and continued advances in positioning systems, e. g., GPS, wireless communication technologies, and electronics in general, renders it increasingly feasible to track and record the changing positions of objects capable of continuous movement. Indexing of such positions is necessary in advanced Location-Based Services (LBS) such as location-based games, tourist-related services, safety-related services, and transport-related services (for example, fleet tracking).

Process Monitoring

Applications such as process monitoring do not depend on positioning technologies. In these, the position of a "moving point" could, for example, be a pair of temperature and pressure values at a specific sensor. A timeslice query would then retrieve all sensors with current measurements of temperature and pressure in given ranges.

Future Directions

Tracking continuous real-world phenomena inevitably involves a high rate of updates that have to be processed by the index. Very recent research provides a number of interesting ideas to speed-up processing of index updates [9,10,22]. Further research is needed to fully explore trade-offs between the update performance and the query performance or the query accuracy. Finally, main-memory indexing of such data could be explored to dramatically boost the performance of index updates.

How to handle the always-present uncertainty regarding the positions of objects has not been sufficiently explored in connection with indexing and warrants further study.

Cross References

► Indexing, BDual Tree
► Indexing of Moving Objects, B^x-Tree
► Indexing, Query and Velocity-Constrained
► Indexing Schemes for Multi-dimensional Moving Objects
► Indexing Spatio-temporal Archives
► Mobile Object Indexing
► Mobile Objects Databases
► Nearest Neighbor Query
► Queries in Spatio-temporal Databases, Time Parameterized
► R*-tree

Recommended Reading

1. Agarwal, P.K., Arge, L., Erickson, J.: Indexing Moving Points. In: Proceedings of the ACM PODS Symposium, pp. 175–186. Dallas, Texas, USA, 14–19 May 2000

2. Basch, J., Guibas, L.J., Hershberger, J.: Data structures for mobile data. In: Proceedings of ACM-SIAM SODA, pp. 747–756. New Orleans, Lousiana, USA, 5–7 Jan 1997
3. Beckmann, N., Kriegel, H.P., Schneider, R., Seeger, B.: The R*-tree: An efficient and robust access method for points and rectangles. In: Proceedings of the ACM SIGMOD Conference, pp. 322–331. Brisbane, Queensland, Australia, 13–16 Aug 1990
4. Benetis, R., Jensen, C.S, Karčiauskas, G., Šaltenis, S.: Nearest and Reverse Nearest Neighbor Queries for Moving Objects. The VLDB Journal **15**(3), 229–249 (2006)
5. Guttman, A.: R-Trees: A Dynamic Index Structure for Spatial Searching. In: Proceedings of the ACM SIGMOD Conference, pp. 47–57. Boston, Massachusetts, USA, 18–21 June 1984
6. Jensen, C.S., Lin, D., Ooi, B.C.: Query and Update Efficient B^+-Tree Based Indexing of Moving Objects. In: Proceedings of the VLDB Conference, pp. 768–779. Toronto, Canada, 31 Aug–3 Sept 2004
7. Jensen, C.S., Tiešytė, D., Tradišauskas, N.: Robust B^+-Tree-Based Indexing of Moving Objects. In: Proceedings of the MDM Conference, p. 12, Nara, Japan, 9–13 May 2006
8. Kollios, G., Gunopulos, D., Tsotras, V.J.: On Indexing Mobile Objects. In: Proceedings of the ACM PODS Symposium, pp. 261–272. Philadelphia, Pennsylvania, USA, 31 May–2 June 1999
9. Lee, M.L., Hsu, W., Jensen, C.S., Cui, Teo, K.L.: Supporting Frequent Updates in R-Trees: A Bottom-Up Approach. In: Proceedings of the VLDB Conference, pp. 608–619. Berlin, Germany, 9–12 Sept 2003
10. Lin, B., Su, J.: Handling frequent updates of moving objects. In: Proceedings of ACM CIKM, pp. 493–500. Bremen, Germany, 31 Oct–5 Nov 2005
11. Mokbel, M.F., Ghanem, T.M., Aref, W.G.: Spatio-temporal Access Methods. IEEE Data Engineering Bulletin **26**(2), 40–49 (2003)
12. Nelson, R.C., Samet, H.: A Consistent Hierarchical Representation for Vector Data. In: Proceedings of the ACM SIGGRAPH Conference, pp. 197–206. Dallas Texas, USA, 18–22 Aug 1986
13. Patel, J.M., Chen, Y., Chakka, Y.P.: STRIPES: An Efficient Index for Predicted Trajectories. In: Proceedings of the ACM SIGMOD Conference, pp. 637–646. Paris, France, 13–18 June 2004
14. Prabhakar, S., Xia, Y., Kalashnikov, D.V., Aref, W.G., Hambrusch, S.E.: Query Indexing and Velocity Constrained Indexing: Scalable Techniques for Continuous Queries on Moving Objects. IEEE Transactions on Computers **51**(10), 1124–1140 (2002)
15. Procopiuc, C.M., Agarwal, P.K., and S. Har-Peled.: STAR-Tree: An Efficient Self-Adjusting Index for Moving Objects. In: Proceedings of the ALENEX Workshop, pp. 178–193, San Francisco, California, USA, 4–5 Jan 2002
16. Šaltenis, S., Jensen, C.S., Leutenegger, S.T., Lopez, M.A.: Indexing the Positions of Continuously Moving Objects. In: Proceedings of the ACM SIGMOD Conference, pp. 331–342. Dallas, Texas, USA, 14–19 May 2000
17. Šaltenis, S., Jensen, C.S.: Indexing of Moving Objects for Location-Based Services. In: Proceedings of ICDE, pp. 463–472. San Jose, California, USA, 26 Feb–1 Mar 2002
18. Samet, H.: The Quadtree and Related Hierarchical Data Structures. Computing Surveys **16**(2), 187–260 (1984)
19. Tayeb, J., Ulusoy, O., Wolfson, O.: A Quadtree Based Dynamic Attribute Indexing Method. The Computer Journal **41**(3), 185–200 (1998)
20. Tao, Y., Papadias, D., Sun, J.: The TPR*-Tree: An Optimized Spatio-temporal Access Method for Predictive Queries. In: Proceedings of the VLDB Conference, pp. 790–801. Berlin, Germany, 9–12 Sept 2003
21. Wolfson, O., Xu, B., Chamberlain, S., Jiang, L.: Moving Objects Databases: Issues and Solutions. In: Proceedings of the SSDBM Conference, pp. 111–122. Capri, Italy 1–3 July 1998
22. Xiong, X., Aref, W.G.: R-trees with Update Memos. In: Proceedings of ICDE, p. 22. Atlanta, Georgia, USA, 3–8 Apr 2006
23. Yiu, M. L., Tao, Y., Mamoulis, Y.: The B^{dual}-Tree: indexing moving objects by space filling curves in the dual space. To appear: VLDB Journal, p. 22 (in press)

Indexing Trajectories

► Movement Patterns in Spatio-temporal Data

Indexing, X-Tree

Daniel Keim[1], Benjamin Bustos[2],
Stefan Berchtold[3], Hans-Peter Kriegel[4]
[1] Computer Science Institute, University of Konstanz, Konstanz, Germany
[2] Department of Computer Science, University of Chile, Santiago, Chile
[3] stb ag, Augsburg, Germany
[4] Institute for Computer Science, Ludwig-Maximilians-University Munich, Munich, Germany

Synonyms

Extended node tree; Access method, high-dimensional; Split, overlap-free; R-tree; Rectangle, minimum bounding; Rectangle, hyper-

Definition

The *X-tree* (eXtended node tree) [1] is a spatial access method [2] that supports efficient query processing for high-dimensional data. It supports not only point data but also extended spatial data. The X-tree provides *overlap-free split* whenever it is possible without allowing the tree to degenerate; otherwise, the X-tree uses extended variable size directory nodes, so-called supernodes. The X-tree may be seen as a hybrid of a linear array-like and a hierarchical R-tree-like directory.

Historical Background

The R-tree [3] and the R*-tree [4], spatial access methods with a hierarchically structured directory that use minimum bounding rectangles (MBRs) as page regions, have primarily been designed for the management of spatially extended, two-dimensional objects, but have also been used for high-dimensional point data. Empirical studies, however, show a deteriorated performance of these spatial access methods for high-dimensional data. The major problem of these index structures in high-dimensional spaces is the overlap between MBRs. In contrast to low-dimensional spaces, there is only a few degrees of freedom for splits in the directory. In fact, in most situations, there is only a single good (overlap-free) split axis. An index structure that does not use this split axis will produce highly overlapping MBRs in the directory and thus show a deteriorated performance. Unfortunately, this specific split axis might lead to unbalanced partitions. In this case, a split should be avoided in order to prevent underfilled nodes.

It is well established that in low-dimensional spaces the most efficient organization of the directory is a hierarchical organization. The reason is that the selectivity in the directory is very high which means that, for example, for point queries, the number of required page accesses directly corresponds to the height of the tree. This, however, is only true if there is no overlap between directory rectangles which is very likely for low-dimensional data. It is also reasonable that for very high dimensionality a linear organization of the directory is more efficient. The reason is that due to the high overlap, most of the directory, if not the whole directory, has to be searched anyway. If the whole directory has to be searched, a linearly organized directory needs less space and may be read much faster from disk than a block-wise reading of the directory. For medium dimensionality, an efficient organization of the directory would probably be partially hierarchical and partially linear. The problem is to dynamically organize the tree such that portions of the data which would produce high overlap are organized linearly and those which can be organized hierarchically without too much overlap are dynamically organized in a hierarchical form.

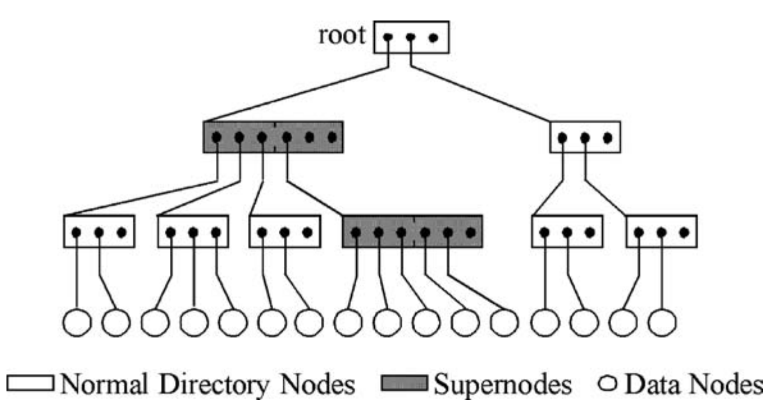
The X-tree is directly designed for the managment of high-dimensional objects and is based on the analysis of problems arising in high-dimensional data spaces. It extends the R*-tree by two concepts: overlap-free split according to a split history and supernodes with an enlarged page capacity. The algorithms used in the X-tree are designed to automatically organize the directory as hierarchically as possible, resulting in a very efficient hybrid organization of the directory.

Scientific Fundamentals

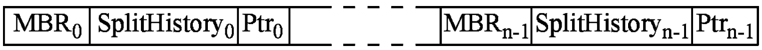
It has been experimentally observed that in high-dimensional spaces a portion of the data space covered by more than one MBR in an R*-tree quickly approaches the whole data space. This is due to the criteria used by the R*-tree to split nodes, which also aims to minimize the volume of the resulting MBRs. The high amount of overlap between MBRs means that, for any similarity query, at least two subtrees must be accessed in almost every directory node, thus reducing the efficiency of the index structure.

To avoid this problem, the X-tree maintains the history of data page splits of a node in a binary tree. The root of the *split history tree* contains the dimension where an overlap-free split is guaranteed (that is a dimension according to which all MBRs in the node have been previously split). When a directory node overflows, this dimension is used to perform the split. However, the overlap-free split may be unbalanced, i. e., one of the nodes may be almost full and the other one may be underfilled, thus decreasing the storage utilization in the directory. The X-tree does not split in this case, but instead creates a *supernode*. A supernode is basically an enlarged directory node which can store more entries than normal nodes. In this way, the unbalanced split is avoided and a good storage utilization is mantained at the cost of diminishing some of the discriminative power of the index.

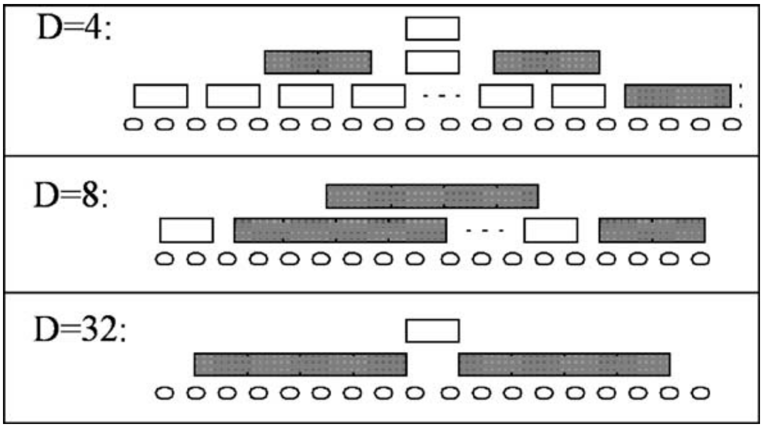
The overall structure of the X-tree is presented in Fig. 1. The data nodes of the X-tree contain rectilinear MBRs together with pointers to the actual data objects, and the directory nodes contain MBRs together with pointers to sub-MBRs (see Fig. 2). The X-tree consists of three different types of nodes: data nodes, normal directory nodes, and supernodes. Supernodes are large directory nodes of vari-



Indexing, X-Tree, Figure 1 Structure of the X-tree



Indexing, X-Tree, Figure 2 Structure of a directory node in the X-tree



Indexing, X-Tree, Figure 3 Various shapes of the X-tree in different dimensions

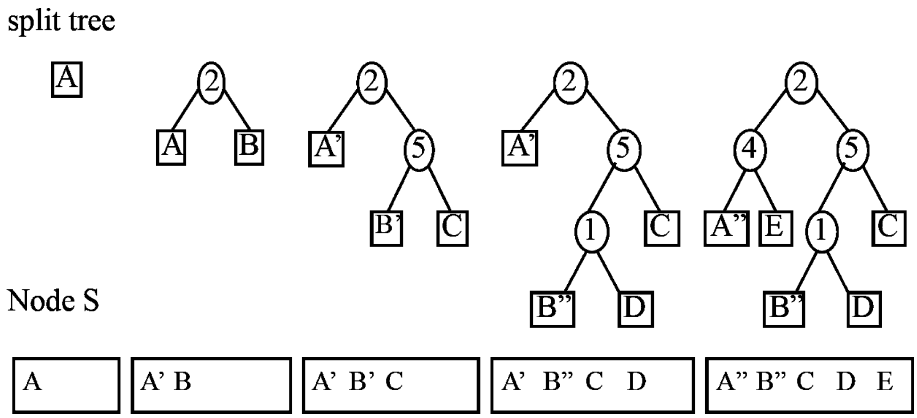
able size (a multiple of the usual block size). The basic goal of supernodes is to avoid splits in the directory that would result in an inefficient directory structure. The alternative to using larger node sizes are highly overlapping directory nodes which would require one to access most of the child nodes during the search process. This, however, is less efficient than linearly scanning the larger supernode.

Note that the X-tree is completely different from an R-tree with a larger block size since the X-tree only consists of larger nodes where actually necessary. As a result, the structure of the X-tree may be rather heterogeneous as indicated in Fig. 1. Due to the fact that the overlap is increasing with the dimension, the internal structure of the X-tree is also changing with increasing dimension. In Fig. 3, three examples of X-trees containing data of different dimensionality are shown. As expected, the number and size of supernodes increases with the dimension. For generating the examples, the block size has been artificially reduced to obtain a drawable fanout. Due to the increasing number and size of supernodes, the height of the X-tree is decreasing with increasing dimension.

The most important algorithm of the X-tree is the insertion algorithm. It determines the structure of the X-tree which is a suitable combination of a hierarchical and a linear structure. The main objective of the algorithm is to avoid splits which would produce overlap. The algorithm first determines the MBR in which to insert the data object and recursively calls the insertion algorithm to actually insert the data object into the corresponding node. If no split occurs in the recursive insert, only the size of the corresponding MBRs has to be updated. In case of a split of the subnode, however, an additional MBR has to be added to the current node which might cause an overflow of the node. In this case, the current node calls the split algorithm which first tries to find a split of the node based on the topological and geometric properties of the MBRs. Topological and geometric properties of the MBRs are, for example, dead-space partitioning, extension of MBRs, etc. The heuristics of the R*-tree [4] split algorithm are an example for a topological split to be used in this step. However, if the topological split results in high overlap, the split algorithm tries next to find an overlap-minimal split which can be determined based on the split history.

For determining an overlap-minimal split of a directory node, one has to find a partitioning of the MBRs in the node into two subsets such that the overlap of the minimum bounding hyperrectangles of the two sets is minimal. In case of point data, it is always possible to find an overlap-free split, but in general it is not possible to guarantee that the two sets are balanced, i. e., have about the same cardinality. It is an interesting observation that an overlap-free split is only possible if there is a dimension according to which all MBRs have been split since otherwise, at least one of the MBRs will span the full range of values in that dimension, resulting in some overlap.

For finding an overlap-free split, a dimension according to which all MBRs of a node S have been previously split has to be determined. The split history provides the necessary information, in particular the dimensions according to which an MBR has been split and which new MBRs have been created by this split. Since a split creates two new MBRs from one, the split history may be represented as a binary tree, called the split tree. Each leaf node of the split tree corresponds to an MBR in S. The internal nodes of the split tree correspond to MBRs which do not exist any more since they have been previously split into new MBRs. Internal nodes of the split tree are labeled by the split axis that has been used; leaf nodes are labeled by the MBR they are related to. All MBRs related to leaves in the left subtree of an internal node have lower values in the



Indexing, X-Tree, Figure 4 Example for the split history

split dimension of the node than the MBRs related to those in the right subtree.

Figure 4 shows an example for the split history of a node *S* and the respective split tree. The process starts with a single MBR *A* corresponding to a split tree which consists of only one leaf node labeled by *A*. For uniformly distributed data, *A* spans the full range of values in all dimensions. The split of *A* using dimension 2 as split axis produces new MBRs *A* and *B*. Note that *A* and *B* are disjointed because any point in MBR *A* has a lower coordinate value in dimension 2 than all points in MBR *B*. The split tree now has one internal node (marked with dimension 2) and two leaf nodes (*A* and *B*). Splitting MBR *B* using dimension 5 as a split axis creates the nodes *B* and *C*. After splitting *B* and *A* again, the situation depicted in the right most tree of Fig. 4 is reached, where *S* is completely filled with the MBRs *A*, *B*, *C*, *D*, and *E*.

One may find an overlap-free split if there is a dimension according to which all MBRs of *S* have been split. To obtain the information according to which dimensions an MBR *X* in *S* has been split, the split tree has to be traversed from the root node to the leaf that corresponds to *X*. For example, MBR *C* has been split according to dimensions 2 and 5 since the path from the root node to the leaf *C* is labeled with 2 and 5. Obviously, all MBRs of the split tree in Fig. 4 have been split according to dimension 2, the split axis used in the root of the split tree. In general, all MBRs in any split tree have one split dimension in common, namely the split axis used in the root node of the split tree.

The partitioning of the MBRs resulting from the overlap-minimal split, however, may result in underfilled nodes which is unacceptable since it leads to a degeneration of the tree and also deteriorates the space utilization. If the number of MBRs in one of the partitions is below a given threshold, the split algorithm terminates without providing a split. In this case, the current node is extended to become a supernode of twice the standard block size. If the same case occurs for an already existing supernode, the supernode is extended by one additional block. Obviously, supernodes are only created or extended if there is no possibility of finding a suitable hierarchical structure of the directory. If a supernode is created or extended, there may not be enough contiguous space on disk to sequentially store the supernode. In this case, the disk manager has to perform a local reorganization.

The algorithms to query the X-tree (point, range, and nearest neighbor queries) are similar to the algorithms used in the R*-tree since only minor changes are necessary in accessing supernodes. The delete and update operations are also simple modifications of the corresponding R*-tree algorithms. The only difference occurs in case of an underflow of a supernode. If the supernode consists of two blocks, it is converted to a normal directory node. Otherwise, that is if the supernode consists of more than two blocks, the size of the supernode is reduced by one block. The update operation can be seen as a combination of a delete and an insert operation and is therefore straightforward.

Key Applications

In many applications, indexing of high-dimensional data has become increasingly important. In multimedia databases, for example, the multimedia objects are usually mapped to feature vectors in some high-dimensional space and queries are processed against a database of those feature vectors [5]. This feature-based approach is taken in many other areas including CAD [6], 3D object databases [7], molecular biology (for the docking of molecules [8]), medicine [9], string matching and sequence alignment [10], and document retrieval [11]. Examples of feature vectors are color histograms [12], shape descriptors [13], Fourier vectors [14], text descriptors [15], etc. In some applications, the mapping process does not yield point objects, but extended spatial objects in a high-dimensional space [16]. In many of the mentioned applications, the databases are very large and consist of millions of data objects with several tens to a few hundreds of dimensions.

Future Directions

The feature-based approach has several advantages compared to other approaches for implementing similarity searches. The extraction of features from the source data is usually fast and easily parametrizable, and metric functions for feature vectors, as the Minkowski distances, can also be efficiently computed. Novel approaches for computing feature vectors from a wide variety of unstructured data are proposed regularly. As in many practical applications where the dimensionality of the obtained feature vectors is high, the X-tree is a valuable tool to perform efficient similarity queries in spatial databases.

Cross References

► Indexing and Mining Time Series Data
► Nearest Neighbor Query
► R*-tree

Recommended Reading

1. Berchtold, S., Keim, D., Kriegel, H.P.: The X-tree: An index structure for high-dimensional data. In: Proc. 22th International Conference on Very Large Databases (VLDB'96), pp. 28–39. Morgan Kaufmann, San Francisco, CA, USA (1996)
2. Böhm, C., Berchtold, S., Keim, D.: Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. ACM Computing Surveys **33**(3), 322–373 (2001)
3. Guttman, A.: R-trees: A dynamic index structure for spatial searching. In: Proc. ACM International Conference on Management of Data (SIGMOD'84), pp 47–57. ACM Press, New York, NY, USA (1994)
4. Beckmann, N., Kriegel, H.-P., Schneider, R., Seeger, B.: The R*-tree: An efficient and robust access method for points and rectangles. In: Proc. ACM International Conference on Management of Data (SIGMOD'90), pp. 322–331. ACM Press, New York, NY, USA (1990)
5. Faloutsos, C.: Searching Multimedia Databases by Content. Kluwer Academic Publishers, Norwell, MA, USA (1996)
6. Berchtold, B., Keim, D., Kriegel, H.-P.: Using extended feature objects for partial similarity retrieval. The VLDB Journal **6**(4), 333–348 (1997)
7. Bustos, B., Keim, D., Saupe, D., Schreck, T., Vranić, D.: Feature-based similarity search in 3D object databases. ACM Computing Surveys **37**(4), 345–387 (2005)
8. Shoichet, B.K., Bodian, D.L., Kuntz, I.D.: Molecular docking using shape descriptors. Journal of Computational Chemistry **13**(3), 380–397 (1992)
9. Keim, D.: Efficient geometry-based similarity search of 3D spatial databases. In: Proc. ACM International Conference on Management of Data (SIGMOD'99), pp. 419–430. ACM Press, New York, NY, USA (1999)
10. Altschul, S. F., Gish, W., Miller, W., Myers, E.W., Lipman D.J.: A basic local alignment search tool. Journal of Molecular Biology **215**(3), 403–410 (1990)
11. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co. Inc., Boston, MA, USA (1999)
12. Shawney, H., Hafner, J.: Efficient Color Histogram Indexing. In: Proc. International Conference on Image Processing, pp. 66–70. IEEE Computer Society, Los Alamitos, CA, USA (1994)
13. Jagadish, H.V.: A retrieval technique for similar shapes. In: Proc. ACM International Conference on Management of Data (SIGMOD'91), pp. 208–217. ACM Press, New York, NY, USA (1991)
14. Wallace, T., Wintz, P.: An efficient three-dimensional aircraft recognition algorithm using normalized fourier descriptors. Computer Graphics and Image Processing **13**, 99–126 (1980)
15. Kukich, K.: Techniques for automatically correcting words in text. ACM Computing Surveys **24**(4), 377–440 (1992)
16. Murase, H., Nayar, S.K.: Three-dimensional object recognition from appearance-parametric eigenspace method. Systems and Computers in Japan **26**(8), 45–54 (1995)

I

Indoor Geolocation

► Channel Modeling and Algorithms for Indoor Positioning

Indoor Localization

MOUSTAFA YOUSSEF
Department of Computer Science,
University of Maryland, College Park, MD, USA

Synonyms

Indoor location determination systems; Tracking; Reference, symbolic; Reference, coordinate based; Position, absolute; Position, relative; FCC 94-102; GPS; Time of flight; Arrival, Time of; Arrival, Angle of; Range Combining; Trilateration; Triangulation; Multilateration; Landmark proximity; WiMAX

Definition

Indoor localization refers to tracking objects in an indoor environment. This tracking can be either in 2-dimensions, 3-dimensions, or 2.5-dimensions. 2.5-dimensions refers to the case when the object position is tracked at discrete plans of the 3-dimensional space, rather than the entire continuum of the 3-dimensional space. For example, tracking a person in multiple 2-dimensional floor-plans in a 3-dimensional building can be considered a 2.5-dimensional tracking.

An indoor location determination system can report the estimated location as a symbolic reference, for exam-

ple, "the lounge", or as a coordinate-based reference. For the coordinate-based reference, the reported tracked-object position can be either relative or absolute. Relative positioning refers to the case where the returned position is relative to a reference point, for example, the x and y coordinates of the position relative to the origin of the map. On the other hand, absolute positioning refers to the case when the returned position is in absolute coordinates, such as the longitude, altitude, and height coordinates.
An indoor location determination system can be centralized or distributed. In a centralized implementation, all the computations are implemented on a centralized server, relieving the computational load from the energy-constrained mobile devices. For a distributed-implementation, the location estimation is performed at the mobile devices. This allows better scalability for the system and, for human tracking, it allows better control over privacy.

Historical Background

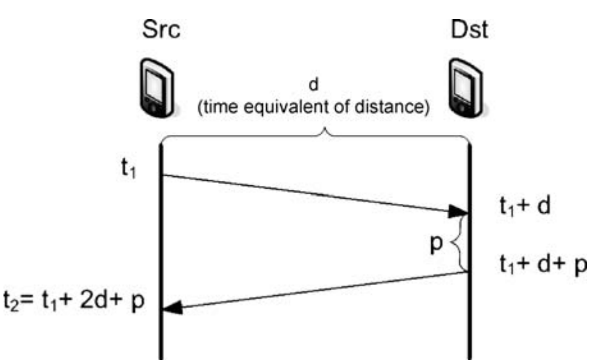
Location determination systems have been an active research area for many years. Since the 1970's, the Global Positioning System (GPS) has been a well known and widely used location determination system. However, the GPS requires a line-of-sight to the satellites and, hence, is not suitable for high-accuracy indoor localization.
Wide-area cellular based systems have been active in developing location determination systems for locating cellular users motivated by the FCC 94-102 order, mandating wireless E911. E911 refers to automatically locating cellular callers who dial the emergency 911 number equivalent to the wired 911 service.
A number of indoor location determination systems have been proposed over the years, including: infrared, ultrasonic, computer vision, and physical contact. All of these technologies share the requirement of specialized hardware, leading to more deployment and maintenance costs and poor scalability.
In the last few years, researchers have started looking at location determination systems that do not require any additional hardware. For example, in an 802.11 WLAN, the wireless card uses the signal strength returned form the access points to roam between different access points. This signal strength information is available to the application level. Therefore, a location determination system based on signal strength information in an 802.11 network, such as the *Horus* system [20], can be implemented without requiring any specialized hardware. This has the advantage of increasing the utility of the data network. A similar idea can be applied to FM radio signals to determine the location of an FM receiver [10] and also to Bluetooth networks [15].

Scientific Fundamentals

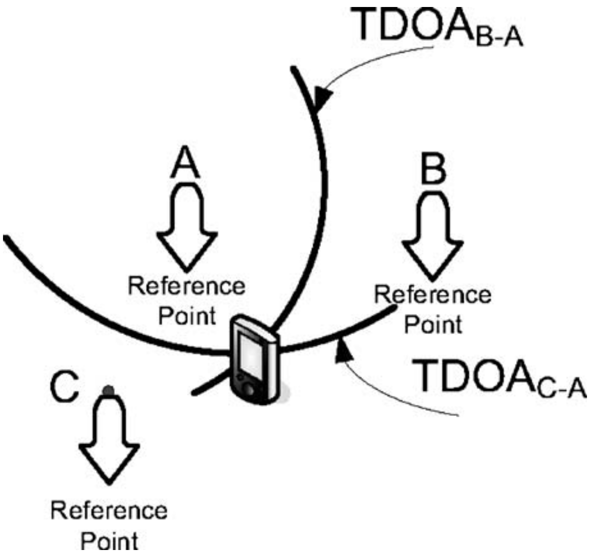
The basic idea used in a location determination system is to measure a physical quantity that is proportional to distance and use the measured value to estimate the distance to a reference point. This process is called ranging. Once the distance is known to one or more reference points, the position of the tracked object can be determined. This process is called range-combining. All location determination systems use these two processes, ranging and range-combining, either explicitly or implicitly. For example, in the GPS system, the reference points are the satellites and the physical quantity used is the time it takes the signal to travel from the satellite to the GPS receiver. The more time it takes the signal to travel from the satellite to the GPS receiver, the larger the distance between them.

Ranging Techniques

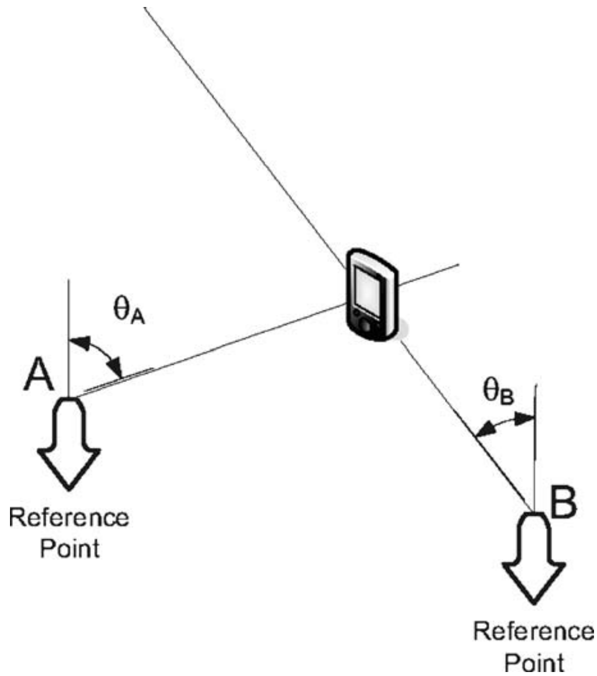
Examples of the signals that can be used in ranging include: Time-of-flight, Angle-of-arrival, and Signal-strength.
Time-of-flight based techniques depend on measuring the time the signal takes to travel from the transmitter to the receiver (called Time-of-Arrival or TOA), the difference of the arrival time at two or more receivers (called Time-Difference-of-Arrival or TDOA), or the time it takes two different signals to reach the receiver from the same transmitter. For example, the system in [13] presented a location technique based on TOA obtained from Round-Trip-Time measurements (Fig. 1) at data link level of the 802.11 protocol. In their system, the sender sends a frame and includes the send timestamp, t_1, in it. As soon as the receiver gets the frame, it sends it back. When the sender gets the frame, it gets a timestamp of the receive time, t_2. The time difference between sending and receiving the frame, $t_2 - t_1$, is used as an estimate of the distance between the



Indoor Localization, Figure 1 The echoing technique for estimating the time of flight. p is the processing delay at the destination. The estimated time is $\frac{t_2 - t_1}{2} = d + \frac{p}{2}$



Indoor Localization, Figure 2 TDOA systems use the principle that the transmitter location can be estimated by intersection of the hyperbolae of the constant differential TOA of the signal at two or more pairs of base stations



Indoor Localization, Figure 3 Based on the estimated AOA of a signal at two or more reference points, the location of the desired unit can be determined

sender and receiver. Note that the processing time established at the receiver affects the accuracy of the estimated distance.

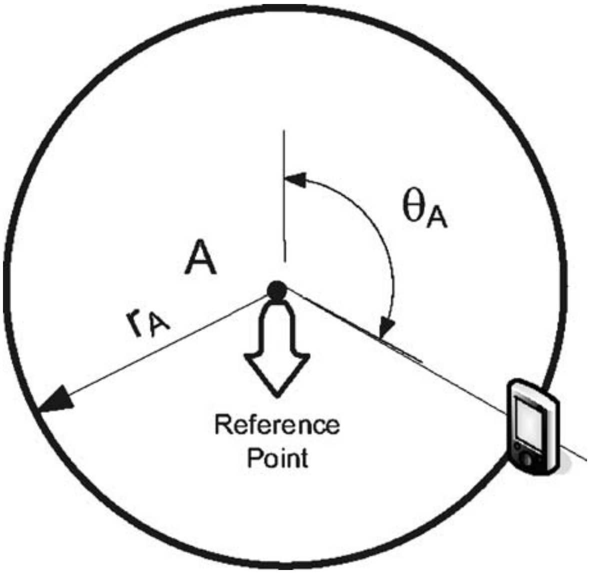
TDOA systems use the principle that the transmitter location can be estimated by the intersection of the hyperbolae of a constant differential TOA of the signal at two or more pairs of base stations (Fig. 2). The idea here is that a message transmitted from the sender is received by three or more receivers. Instead of keeping track of the absolute TOA of the signal, TDOA-based systems keep track of the difference of reception times of the transmitted message at the different receivers. Given two receivers' locations and a known TDOA, the locus of the sender location is a hyperboloid. For more than two receivers, the intersection of the hyperbolae associated with the TDOA of each pair of receivers provides the final transmitter's location.

Systems that use two different physical signals, e. g., the Cricket System [16], which uses ultrasound and RF signals, use one of the signals for synchronization and the other for the time estimation. The idea is that the speed of the ultrasonic signal is much lower than the speed of the RF signal. Therefore, when the sender transmits an RF signal followed by an ultrasound signal, the receiver can use the difference in time between the reception of the ultrasound signal and the RF signal as an estimate of the distance between the sender and the receiver since the time it takes the RF signal to reach the receiver is negligible compared to the time it takes the ultrasound signal to reach the same receiver.

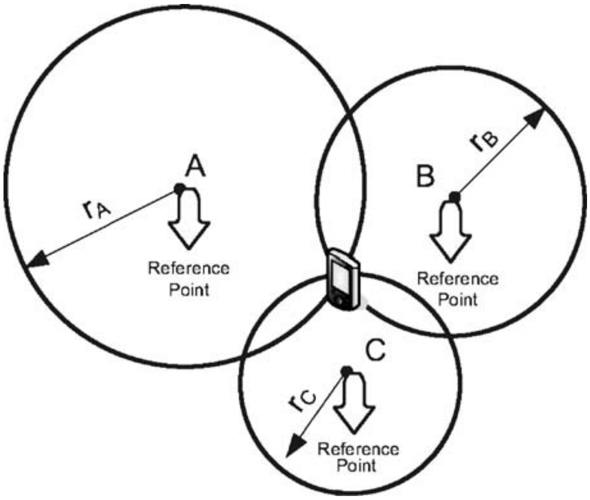
Angle-of-arrival (AOA) based techniques use antenna arrays to estimate the angle of arrival of the signal from the transmitter. The idea is to measure the difference of arrival time of the signal at individual elements of the array and use the delay to estimate the AOA. Based on the estimated AOA of a signal at two or more reference points, the location of the desired unit can be determined as the intersection of a number of lines (Fig. 3).

Signal-strength based techniques, e. g., the *Horus* system, use the signal strength received from a reference point as an estimate of how close the reference point is. For outdoor environments, the relation between signal strength and distance can be approximated by a logarithmic function. However, for indoor environments, this relation is very complex due to multiple phenomena, such as the multipath effect and signal diffraction, among others. Therefore, indoor location determination systems that use signal strength usually use a lookup table to store the relation between the signal strength and distance. This table has been called a "radio-map" in the literature.

An example of another possibility that implies an implicit measurement of a physical quantity is the Cell-ID based method. In **Cell-ID** based methods, e. g., RF-IDs, the location of the transmitter is taken to be the location of the nearest base station it is associated with.



Indoor Localization, Figure 4 Combining propagation time measurement with angle measurement to obtain the position estimate can be done by using only one reference point



Indoor Localization, Figure 5 Trilateration locates a node by calculating the intersection of three circles. If the ranges contain an error, the intersection of the three circles may not be a single point

Hybrid methods can be used that combine two or more of these techniques. For example, combining propagation time measurement with angle measurement to obtain the position estimate can be done by using only one reference point (Fig. 4).

To obtain these physical measurements, different underlying communication technologies can be used. This includes infrared, ultrasonic, radio frequency (RF), computer vision, and physical contact, among others.

Range Combining Techniques

Once the range information is gathered from a number of reference points, it needs to be combined to estimate the location. This process is called Range-combining. Trilateration, triangulation, multilateration, and landmark-proximity are the most common techniques for combining ranges.

Trilateration refers to locating a node by calculating the intersection of three circles (Fig. 5). Each circle is centered at a reference point with a radius equal to the estimated range between the reference point and the node. If the ranges contain errors, the intersection of the three circles may not be a single point.

Triangulation is used when the angle of the node, instead of the distance, is estimated, as in AOA methods. The nodes' positions are calculated in this case by using the trigonometry laws of sines and cosines. In this case, at least two angles are required.

In **multilateration**, the position is estimated from distances to three or more known nodes by minimizing the error between the estimated position and the actual position by solving a set of non-linear equations.

Proximity-based techniques are usually used when no range information is available. For example, the GPS-less system [6] employs a grid of beacon nodes with known locations; each unknown node sets its position to the centroid of the beacon locations it is connected to.

Key Applications

Indoor localization can be used in many applications, most notably in context-aware applications and enhancing network protocols.

Context-Aware Applications

The context of an application refers to the information that is part of its operating environment. Typically, this includes information such as location, activity of people, and the state of other devices. Algorithms and techniques that allow an application to be aware of the location of a device on a map of the environment are a prerequisite for many of these applications. Examples of location-aware applications [7,8,18] include location-sensitive content delivery, where tailored information is sent to the user based on his current location, direction finding, asset tracking, teleporting, robotics, and emergency notification.

Asset Tracking Location information can be used to track assets in indoor environments. For example, RF-IDs have been widely used for tracking assets in military and civilian applications. Note that the technologies that can be used for asset tracking can also be used for tracking humans, such as in [19].

Direction Finding Another important application for indoor localization is to find the direction and route between two points. This is similar to the GPS-based navigation systems in cars today, however, it is applied to indoor environments. For example, in the Shopping Assistance system [3], the device can guide the shoppers through the store, provide details of items, help locate items, point out items on sale, do a comparative price analysis, and so forth. There is a privacy concern since the store maintains the customer profiles. As a consequence, customers are divided into two classes. The first class is the regular customers who shop anonymously without profiles in the store. The second class is the store customers who signed up with a store will get additional discounts in exchange for sacrificing their privacy.

Guided Tours In guided tour applications, e. g., [2], information can be displayed on a device carried by a user based on the device's current location. The user can also leave comments on an interactive map. This kind of tailored information enhances the user experience.

Call Forwarding In this application[17], based on the Active Badge System, the user location is tracked in a central server that is connected to the enterpriser phone system. Whenever a call arrives to a user who is not currently in his office, the call is automatically routed to the room the user is located based on his/her current location.

Teleporting The Teleporting System [4], developed at the Olivetti Research Laboratory (ORL), is a tool for experiencing obile applications The system allows users to dynamically change the display device from which their currently running applications are accessible. It operates within the X Window System and allows users to interact with their existing X applications at any X display within a building. The process of controlling the interface to the teleporting system comes from the use of an automatically maintained database of the location of equipment and people within the building.

Robotics Finding the location of robots in indoor environments is crucial in many applications. For example, the system in [11] uses radio frequency identification (RFID) for robot-assisted indoor navigation for the visually impaired. Robots equipped with RFIDs and laser range finders allow visually impaired individuals to navigate in unfamiliar indoor environments and interact with the robotic guide via speech, sound, and wearable keyboards.

Network Protocols Enhancements

The second class of applications for indoor location determination systems is enhancements for network protocols. This usually applies to sensor network applications for indoor environments. These enhancements include determining the location of an event, location-based routing, node identification, and node coverage.

Determining the Location of an Event Determining the location of an event is an important service that is particularly important in indoor sensor networks [5]. In indoor sensor networks, it is always important to record the location of an event whenever the event occurs. This highlights the importance of indoor location determination systems in such applications.

Location-Based Routing A number of location based routing protocols have been proposed for using the location information of the sender and receiver to achieve scalable routing protocols. For example, the GPSR protocol [9] exploits the correspondence between geographic position and connectivity in a wireless network by using the positions of nodes to make packet forwarding decisions compared to the standard routing protocols that use graph-theoretic notions of shortest paths and transitive reachability in order to find routes. GPSR uses greedy forwarding to forward packets to nodes that are always progressively closer to the destination. In regions of the network where such a greedy path does not exist (i. e., the only path requires that one move temporarily farther away from the destination), GPSR recovers by forwarding in perimeter mode, in which a packet traverses successively closer faces of a planar subgraph of the full radio network connectivity graph until reaching a node closer to the destination where greedy forwarding resumes.

Node ID The inspection of building structures, especially bridges, is currently made by visual inspection [12]. The few non-visual methodologies make use of wired sensor networks which are relatively expensive, vulnerable to damage, and time consuming to install. Recordings of structures during ambient vibrations and seismic disturbances are essential in determining the demand placed upon those structures. For structures in high seismic areas, information provided by monitoring structural responses will inevitably lead to a better scientific understanding of how structures behave in the nonlinear realm. Using structure monitoring sensor networks is vital in these environments. It is a challenge for such huge indoor sensor networks to determine the node IDs for a large number of randomly placed nodes. Location determination can be used as node identification in these environments where the node location is used as its ID.

Node Coverage One of the fundamental issues that arises in sensor networks is coverage. Coverage can be considered as the measure of quality of service of a sensor network [14]. For example, in a fire detection sensor network scenario, one may ask how well the network can observe a given area and what the chances are that a fire starting in a specific location will be detected in a given time frame. Furthermore, coverage formulations can try to find weak points in a sensor field and suggest future deployment or reconfiguration schemes for improving the overall quality of service. For the coverage problem, knowing the nodes' locations is essential for protocols that address this problem.

Future Directions

New different technologies, e. g., WiMax, are being developed that will allow larger transmission ranges and more accurate measurements of the physical quantities. This should allow more accurate and ubiquitous localization. As more accurate localization techniques are being introduces, a new set of applications are emerging to take advantage of these localization capabilities, including GPS-less city wide localization [1].

Cross References

- ► Channel Modeling and Algorithms for Indoor Positioning
- ► Indoor Positioning, Bayesian Methods
- ► Radio Frequency Identification (RFID)

Recommended Reading

1. http://www.skyhookwireless.com/
2. Abowd, G.D., Atkeson, C.G., Hong, J., Long, S., Kooper, R., Pinkerton, M.: Cyberguide: A mobile context-aware tour guide. Wireless Networks **3**(5), 421–433 (1997)
3. Asthana, A., Cravatts, M., Krzyzanowski, P.: An indoor wireless system for personalized shopping assistance. In: IEEE Workshop on Mobile Computing Systems and Applications, pp. 69–74. Santa Cruz, California (1994)
4. Bennett, F., Richardson, T., Harter, A.: Teleporting - making applications mobile. In: IEEE Workshop on Mobile Computing Systems and Applications, pp. 82–84. Santa Cruz, California (1994)
5. Brooks, A., Makarenko, A., Kaupp, T., Williams, S., Durrant-Whyte, H.: Implementation of an indoor active sensor network. In: Int. Symp. on Exp. Robotics, Singapore (2004)
6. Bulusu, N., Heidemann, J., Estrin, D.: GPS-less Low-cost Outdoor Localization for Very Small Devices. In: IEEE Personal Communications (2000)
7. Chen, G., Kotz, D.: A survey of context-aware mobile computing research. Technical Report Dartmouth Computer Science Technical Report TR2000–381 (2000)
8. Gellersen, H.-W., Schmidt, A., Beigl, M.: Adding some smartness to devices and everyday things. In: IEEE Workshop on Mobile Computing Systems and Applications, pp. 3–10. December (2000)
9. Karp, B., Kung, H.T.: Greedy perimeter stateless routing for wireless networks. In: Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000), pp. 243–254: Boston, MA (2004)
10. Krumm, J., Cermak, G., Horvitz, E.: Rightspot: A novel sense of location for a smart personal object. In: Fifth International Conference on Ubiquitous Computing, October (2003)
11. Kulyukin, V., Gharpure, C., Nicholson, J., Pavithran, S.: RFID in robot-assisted indoor navigation for the visually impaired. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004, pp. 1979–1984. (2004)
12. Lynch, J.P., Kiremidjian, A.S., Law, K.H., Kenny, T., Carryer, E.: Issues in wireless structural damage monitoring technologies. In: 3rd World Conference on Structural Control (WCSC), Como, Italy (2002)
13. Sebastiano, C., Ciurana, M., Barcelo,F.: Indoor tracking in wlan location with toa measurements. In: MobiWac 2006 (2006)
14. Meguerdichian, S., Koushanfar, F., Potkonjak, M., Srivastava, M.B.: Coverage problems in wireless ad-hoc sensor networks. In: INFOCOM, pp. 1380–1387 (2001)
15. Peddemors, A.J.H., Lankhorst, M.M., de Heer, J.: Presence, Location, and Instant Messaging in a Context-Aware Application Framework. Proc. of the Intl. Conf. on Mobile Data Management (MDM) **2574**, 325–330
16. Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The cricket location-support system. In: 6th ACM MOBICOM, Boston, MA, August (2000)
17. Want, R., Hopper, A., Falcão, V., Gibbons, J.: The active badge location system. ACM Transactions on Information Systems **10**(1), 91–102 (1992)
18. Want, R., Schilit, B.: Expanding the horizons of location-aware computing. IEEE Computer **34**(8), 31–34 (2001)
19. Want, R., Schilit, B.N., Adams, N.I., Gold, R., Petersen, K., Goldberg, D., Ellis, J.R., Weiser, M.: An overview of the parctab ubiquitous computing experiment. IEEE Personal Communications **2**(6), 28–43 (1995)
20. Youssef, M., Agrawala, A.: The Horus WLAN Location Determination System. In: Third International Conference on Mobile Systems, Applications, and Services (MobiSys 2005), June (2005)

Indoor Location Determination Systems

► Indoor Localization

Indoor Location Estimation

► Channel Modeling and Algorithms for Indoor Positioning

Indoor Position Estimation

► Channel Modeling and Algorithms for Indoor Positioning

Indoor Positioning

CLAUDE CARON[1],
DANIEL CHAMBERLAND-TREMBLAY[1],
CÉDRIC LAPIERRE[1], PIERRE HADAYA[3],
STÉPHANE ROCHE[2], MOKHTAR SAADA[1]
[1] GeoBusiness Group, Faculty of Administration, University of Sherbrooke, Sherbrooke, QC, Canada
[2] Center for Research in Geomatics, Pavillon Louis-Jacques-Casault, University of Laval, Québec, QC, Canada
[3] Department of Management and Technology, University of Québec at Montréal, Montréal, QC, Canada

Synonyms

Indoor positioning system; Microgeomatics; Real-time location services; Location-based services; Spatial statistical analysis; Smart buildings

Definition

Indoor positioning is a technique that provides the continuous real-time location of objects or people within a closed space through measurements [1]. It is primarily used in retail floors, warehouses, factories, and offices to monitor and track people, equipments, merchandise, etc. Contrary to self-positioning systems such as GPS, indoor positions are calculated on a distant server using the information transmitted by mobile tags [2]. Indoor positioning systems (IPS) may have different configurations. For example, tags may transmit movement information directly through a wireless network or may be read by scanners as they pass by. In the latter case, tags possess no processing capabilities and are unable to calculate their own position. Also, different radio frequencies can be used for indoor positioning: namely, ultrasound, a wireless LAN-based signal such as wireless fidelity (WiFi) or Bluetooth, and cellular network signals. Finally, to be completed, the process of indoor positioning usually requires data analysis (e. g., validation, spatial analysis, geostatistics, data mining) to extract patterns and trends as well as to provide accurate and timely knowledge to managers.

Historical Background

Dedicated IPS have emerged from the need to have accurate repeated location measurements of tangibles such as humans and equipments. Hightower and Borrielo [3] trace back the origin of these systems to the "Active Badge" project [4] which aimed at replacing the traditional "call and report back" pager approach to locate people.

Weiser [5,6] was one of the first to recognize the importance and potential of indoor location and viewed it as a central component of ubiquitous computing. In articulating his vision of the next generation computing, he identified three innovations that would transform computer science [7]:

- Computing devices will become embedded in everyday objects and places.
- Designers will develop intuitive, intelligent interfaces for computing devices to make them simple and unobtrusive for users.
- Communications networks will connect these devices together and will evolve to become available anywhere and anytime.

In time, development efforts in the field of indoor location technologies echoed these principles. Today, the field offers great opportunities for context-aware computing, whether ubiquitous, pervasive or mobile (for examples, please refer to Future Directions, below).

Indoor positioning also borrows from the great tradition of land surveying. Over the last 20 years, this field of research has embraced computer science and new technologies to transform itself into what is known nowadays as geographic information sciences (GISciences). The focus of GISciences has broadened from geocentric measurements (land registry, natural resources and public utilities management) to include novel application domains such as business (e. g., geomarketing, fleet management), communication (e. g., peer finder) and sports (e. g., geocaching, precision training). Evidently, information technology innovations such as wireless communication (e. g., Wi-Fi, WiMax), mobile devices [e. g., presonal digital assistants (PDAs), mobile phones] and new location techniques [e. g., assisted global positioning system (A-GPS) on mobile phones] are now important components of GISciences. Today, indoor positioning is able to take advantage of the topological analysis and mapping expertise found in GISciences.

Indoor positioning did not evolve from GISciences, however. Devices emerged from various technical fields. Indeed, Beresford [8] provides examples of mechanical, magnetic, acoustic, radio and microwave, optical, and inertial systems. Of these, radio-frequency-based systems, namely radio frequency identification (RFID), attracts most attention among scholars and practitioners alike. New applications of RFID technologies are appearing rapidly and have profound impacts on processes and strategies of organizations (Batty 2004). RFID technology appeared in the 1940s, and was initially used mainly for military purposes [9] but in the 1990s, new standards were developed and they contributed to the deployment of the technology in numerous other contexts. In contrast to GPS, the RFID approach is foreign to the GISciences. It is rooted in the field of supply chain management (SCM) and viewed as a powerful replacement for barcode systems. The integration of RFID into spatial analysis and geostatistics has led the way to a new domain of expertise: “microgeomatics” [10]. Microgeomatics is the field of real-time tracking and spatial analysis of people and equipment motions in a closed space, such as merchandise in a warehouse (e. g., Walmart), customers in a retail store, airline luggage, family pets and medical equipments. Such applications are currently deployed in an increasing number of organizations [11], where important financial investments and operational adjustments are generally required. Indoor positioning capabilities play a highly strategic role for businesses. The best strategic fit appears where the appropriate identification and location data can be integrated to the management systems such as ERP, CRM and SCM.

Scientific Fundamentals

In order to properly understand the scientific fundamentals of indoor positioning, this section will highlight the telecommunication methods, describe the two dominant positioning methods, and) detail the positioning techniques adopted in the field.

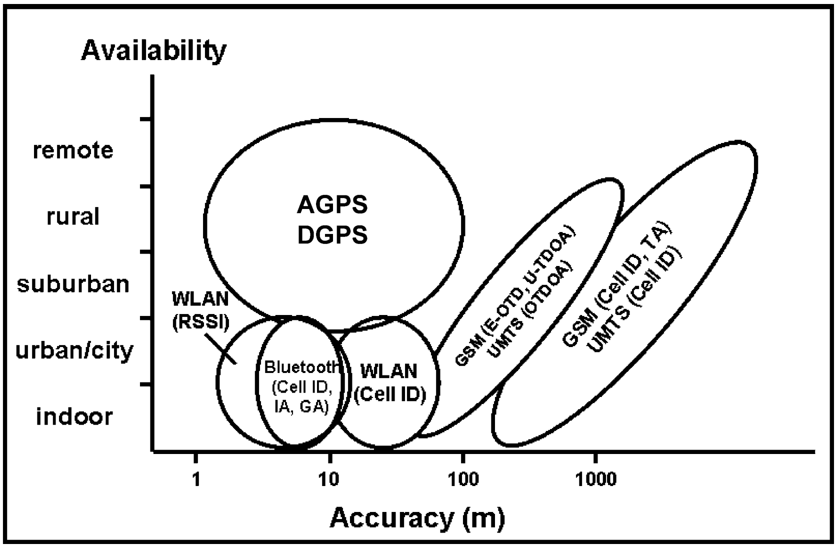
Telecommunication Methods

Various telecommunication methods can be used by location systems: radio frequencies [RFID/Wi-Fi/cellular/Bluetooth/ultrawideband (UWB)], infrared radiation, ultrasound, optics and electromagnetic fields to name a few [12,13]. The use of radio frequencies remains, however, the most popular telecommunication approach adopted for indoor positioning because of its ease of deployment and low cost. Even though field studies have demonstrated that any radio frequency (RF) can be used for indoor positioning, the choice of radio frequencies should vary according to the accuracy requirement and the positioning environment (Fig. 1). In the context of indoor systems, numerous frequencies can be used: Wi-Fi (IEEE 802.11), bluetooth (IEEE 802.15), wide-area cellular and GPS/UWB [12].

Two Dominant Indoor Positioning Methods

The RFID Method An RFID-based system is “an emerging technology intended to identify, track, and trace items automatically” [15]. This positioning method is claimed to add intelligence and minimize human intervention in the identification process by using electronic tags. An RFID application is comprised, at a minimum, of four components: tags, antennae, readers and software.

An RFID tag is a transponder usually placed on objects to identify their locations. It is made of a coil, an antenna and a microchip. One way of classifying tags is to divide them into passive and active tags. Passive tags are relatively cheap, have a short range of detection, are read-only, and are powered by remote antenna signals. In contrast, active tags [often called real-



Indoor Positioning, Figure 1 Accuracy per positioning environment [14]. *AGPS*, Assisted Global Positioning System; *DGPS*, Differential Global Positioning System; *WLAN*, Wireless Local Area Network; *Cell ID*, Cell Identification; *IA*, Incremental Algorithm; *GA*, Geometric Algorithm; *GSM*, Global System for Mobile communications; *E-OTD*, Enhanced Observed Time Difference; *U-TDOA*, Uplink Time Difference of Arival; *UMTS*, Universal Mobile Telecommunications System; *OTDOA*, Observed Time Difference of Arival; *TA*, Timing Advance based positioning; *RSSI*, Received Signal Strength Indicator time location systems (RTLS)], are more expensive, have a longer read/write range, offer greater functionality and must be battery powered. The battery of the active tag allows for repeated and autonomous transmissions of radio waves used for positioning; its life span is usually up to 5 years.

Passive RFID tags are slowly replacing barcodes as they become less expensive. They provide automatic contactless capture of information and do not require line-of-sight to work. RFID tags, passive and active alike, are used to manage various processes (e. g., warehouse, logistics, access control) in numerous industries (e. g., agriculture, healthcare) [9].

In recent years, researchers have tackled the problem of real-time positioning using RFID passive and/or active tags. Two approaches have been developed [16]. The first one requires fixing the tags on objects inside a room and placing a short-range receiver on the person or object in motion. This technique is constrained by the size of the antenna and the power requirement, thus limiting receivers to short-range models. The second method requires installing powerful antennae in a room to create zones and attaching tags to the objects or persons in motion.

Antennae come in many shapes and forms, and present different technical characteristics. They vary in size from less than a square centimeter to several square meters. Ultrahigh frequency (UHF) reader antennae can be classified as circular-polarized or linear-polarized antennae [15]. Circular-polarized antennae emit and receive radio waves from all directions, are less sensitive to transmitter-receiver orientation and work better "around corners". Linear-polarized antennae work in one particular direction but have a longer range.

Readers are used to query or read all the tags within their range in quick succession. In interrogation, the reader sends a signal to the tag and listens. In reading, active tags continually send out signals to the reader. To read passive tags, the reader sends them radio waves which energize the tags as they start broadcasting their data [15]. The range of passive tags increases every year. It was less than 1 m a few years ago and today it can be up to 10 m [12]. A reader's range rests on its transmission power, the size of its antenna, the size of the antenna of the tag and its relative orientation, as well as on he presence of metal structures in its surroundings [16].

The software components are responsible for the integration of an RFID system. In a typical setting, a software front-end component manages the readers and the antennae while a middleware component routes this information to servers running the backbone database applications [15].

The Wi-Fi Method Many working environments and shopping centers provide their users with wireless high-speed internet access. The infrastructure required to support these connections is affordable, easily deployed and user oriented. The wireless infrastructure transmits at 2.4 GHz and has a reach of 50–100 m depending on the environment and the transfer rate [12].

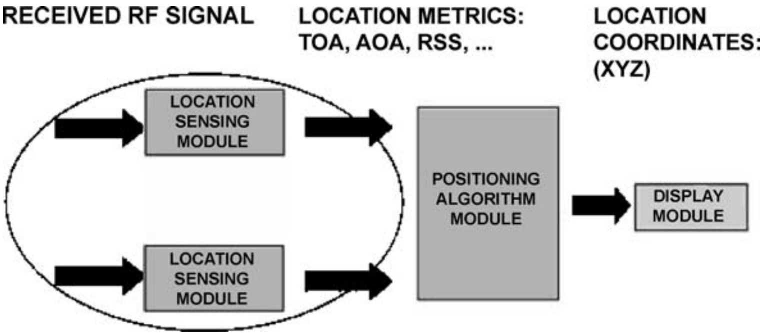
The Wi-Fi location method takes advantage of existing wireless infrastructures present in many buildings. However, in using radio-waves-based positioning, it is imperative to take into account the errors, distortions and inaccuracies tied to the propagation of the signal as obstacles are encountered and longer distances are covered. Indeed, according to experts, the propagation of the signal varies from site to site and cannot be modeled in a general fashion [13,17,18,19].

Reflection, diffraction and dispersion of radio waves are the main causes of measure distortion, signal loss and multipath effect. Movements of object, temperature variations and air displacement also account for random irregularities in the propagation of the signal causing positioning inaccuracy. The radio signal strength (RSS) metric is further affected by electronic devices operating on the WiFi band causing lost signal areas, noise and interferences [13,20]. This latter problem is important since the frequency is open to all manufacturers of microwaves, cordless phones and Bluetooth devices. Finally, the orientation of the person or object carrying the location device has an influence on the strength of the measured signals; the person or object can block, partially or entirely, the signal arriving at certain access points. For example, the human body, comprised mainly of water, absorbs waves at 2.4 GHz and has a direct impact on the reception of the Wi-Fi signal [20].

Positioning Techniques

According to Pahlavan et al. [17], the process of indoor positioning is initiated by the reception of the RF signal of a tag by sensing modules (antennae) (Fig. 2). These modules have a known and fixed location. They measure specific metrics of the tag transmission such as RSS, time of arrival (TOA) or angle of arrival (AOA) and relay the measurements to the positioning module. The measurements are then pooled and used to calculate the position of the tag. Finally, these positions are sent to be visualized by a user or can be stored in a database for archival and future analyses.

Hightower and Borriello [3,21] classify positioning techniques into four categories: geometric triangulation (including lateration and angulation methods), statistical analysis, proximity, and scene analysis.



Indoor Positioning, Figure 2 Indoor positioning process [17]. *TOA* Time of arrival, *AOA* angle of arrival, *RSS* radio signal strength, *RF* radio frequency

Geometric Technique This technique uses a geometric algorithm based on the trilateration method or the triangulation approach. In the trilateration method, a position is found at the intersection of the arcs defined around the network antennae by the distance of the detected object. The distance is computed based on the travel time of a signal from the source to the receiver (e. g., TOA/TDOA). This approach is based on a fixed and often costly infrastructure that guarantees the necessary synchronization and precision [17,22]. Distance can also be estimated with a correlation function between the signal attenuation (RSS) and the distance between the source and the receiver [22,23].

The triangulation approach is based on a known distance between two or more antennae and the corresponding angles of an incoming signal to position an object or a person. This approach requires the use of directional antennae and the knowledge of the distance between antennae on a line [22]. Measurements based on signal propagation depend upon a constant signal. Indoor environments seldom offer such stability; radio waves vary widely as they often collide with objects such as walls, furniture, equipments and other obstacles. Special infrastructures are used to circumvent this problem and obtain reliable metrics. Still, the geometric technique often yields coarse areas rather than exact point locations [17]. In such cases, the use of direct or iterative positioning algorithms by the system may help to gain better precision.

Statistical Analysis This technique, also called "empirical method," uses the "pattern matching algorithm" [13]. It relies on metrics such as RSS to create the "location fingerprint" of a tag [23]. Statistical analysis uses the RSS value directly for positioning and not as an input in a calculation as is the case with the geometric technique. According to Pahlavan et al. [17], every area in a building has a unique signature (RSS) tied to the signal propagation and location of antennae. During a preliminary benchmarking phase, the system records all the signatures of these areas in a database. These signatures are then used by different algorithms to acknowledge the position in operational mode [13]. Next, the receiver receives the signals from the antennae at a predetermined number of seconds and the information is sent back in the network. The server processes the signals in a filter that makes an average according to the delay determined (number of seconds) prior to going through the deterministic and probabilistic algorithms [14,17,18,19,22].

The advantage of this method lies in that the exact knowledge of the positions of the antennae is not required. The statistical analysis method is particularly appropriate for indoor environments as it can take into account the signal obstruction caused by different objects when calculating the location. However, the method suffers from the potentially long benchmarking process, especially for extensive indoor space coverage, which needs to be repeated every time the radio environment changes.

Proximity Technique This technique, also called "cell-ID based positioning" or "cell of origin", is based on the proximity of detectable objects to a point of reference[3,21]. It includes both indoor and outdoor positioning technologies that fall into three categories: (1) Physical contact to a sensor (e. g., pressure, capacity, movement), (2) inference from an identification system (e. g., credit card in a retail store), and (3) connection to a network access point of known radius of action (e. g., traditional cellular phone location). More specifically in the latter category, the position of the source of a signal corresponds to the zone of the receiver (antenna) coverage. Thus, the precision of a location varies according to the size of the zone covered. The size of a detection zone can be adjusted by varying the strength of the signal of the receiver. By lowering the signal strength, zones become smaller and precision increases.

The proximity technique is suitable for a large number of applications such as manufacturing, access control, healthcare management, industrial maintenance, container tracking, computer hardware management and tracking of personnel.

Indoor Positioning, Table 1 Categories of indoor LBS applications and their characteristics

Service category	Example application	Characteristics	
		Telecommunication method	Indoor positioning method and technique
Infotainment services	In a ubiquitous tourism perspective, use urban Wi-Fi or cellular network to access information concerning relevant local events in the surroundings	Wi-Fi (WiMax), wide-area cellular and AGPS	Wi-Fi, proximity technique
Tracking services	In a business-to-business perspective, place and track automatically an order to a hardware supplier, by billing and managing warehouse inventory through RFID technology	Wi-Fi, WLAN	RFID, statistical analysis
Selective information dissemination	In an hospital, allow for the transmission of the patient file on PDA (personal digital assistant) only to nurses entering the patient's room; control the access of hospital restricted areas	Bluetooth, Wi-Fi	Wi-Fi, geometric technique, scene analysis
Location-based games	With Web-based games, propose a default setup (language, country, etc.) according to the location of the gamer (IP address, cellular zone)	Cellular,GPS/UWB, IP address	Wi-Fi, proximity technique
Emergency support services	Upon a fire alert, provide automatically the detailed building map on PDA to the firefighter, indicate life-threatening situations, and track him in risk areas (e.#8239;g., toxic products)	Wi-Fi, Wide-Area Cellular, AGPS	Wi-Fi, geometric technique
Location-sensitive billing	Automatic RFID tag reading for continuous flow through highway tolls, an invoice is sent periodically by mail	RFID/Wireless (Wi-Fi) or Bluetooth	RFID, proximity technique

Scene Analysis A scene analysis system identifies and locates users inside a building using special colored badges (visual tags) and video analysis. This technique requires the use of a network of cameras equipped to detect the tags in their video streams. The location of a tag can be detected by performing a triangulation when two or more cameras detect the same tag. This is possible because: (1) the size of the badges is fixed (2) the positions of the cameras are known, and (3) the angle of the cameras is known when detecting a tag [24].

A similar approach relies on image analyses for shape recognition. This technique has been used to detect and locate pieces of equipment and people, using their physical traits such as the shape of their face or the clothes they wear, in an assembly line.

Key Applications

Indoor positioning is a special case of the application field of location-based services (LBS). It has evolved to complement GPS positioning which is almost exclusively restricted to outdoor environments.

A LBS system is generally defined as any system providing a value-added service based on the location of a mobile device in a wireless environment. It can be viewed as a collection of intelligent agents capable of initiating a communication to obtain services based on the state or activity of a user or a device. The applications are based on the integration of wireless technology, computer science, positioning technologies and spatial data [25]. LBS applications deployed and used indoors are specifically considered indoor positioning applications.

Different configurations of LBS applications are possible. Some applications send information directly to the user when the person's position coincides with some predetermined or user-defined parameters. Other applications only work when manually activated by the user. Based on this differentiation criterion, three models of LBS applications are recognized:

- Pull-type or "user-requested" applications: the user requests a service at a point in time based on the user's location.
- Push-type or "triggered" applications: a service is provided automatically based on the location of a user as soon as a set of predefined conditions are satisfied. In this case, the system collects the position of the device to provide the service.
- Tracking applications: the system provides information about a mobile device (e. g., cellular phone) upon request from a user. Services based on this model combine the "push" and "pull" approaches.

Some authors also distinguish LBS applications according to whether the services are provided to a person or a device. In the case of people, the application uses the position of a user to provide a value-added service. For devices, it is the position of the device or the equipment that triggers the application (e. g., the arrival of merchandise in a warehouse triggers an update of the inventory database).

Jacobsen [24] classifies LBS into six categories. Table 1 presents an example of some applications, as well as their particular characteristics for each of these six categories.

Given the infancy of the field, some indoor positioning applications mentioned in Table 1 are already considered mature, whereas others still are at the forefront of research and development. Numerous applications, such as identification and tracking of individuals through face recognition for security purposes or pathway analysis of customer behavior in shopping centers, are already available today.

Many specialists recognize the great potential of application development for indoor positioning. However, the increasing number of technological opportunities makes understanding the application market an ever more complex task. As indoor positioning technologies become common place, it will be easier to capture the market and identify customers' needs as well as application niches.

Future Directions

Despite all the advances in the field, the diffusion of indoor positioning is still limited considering its potential. An increasing number of research, development, and business projects will appear in the next few years. Future works will spread beyond technological boundaries into four major and complementary research streams: technology performance and evolution, organizational benefits, individual benefits, and people and society.

Technology Performance and Evolution

Indoor positioning technologies are benefiting from the ongoing research and development efforts in the fields of telecommunications (e. g., 3G) and RF analysis. Systems will gain better accuracy and precision and increase their range of operation as new approaches and algorithms are being developed (e. g., fuzzy logic, neural networks, hidden Markov) [17]. In addition, turnkey solutions will be commercialized as standards emerge and facilitate the integration of indoor positioning applications with corporate technology infrastructures. As microgeomatics and GISciences influence each other, new dedicated analytical tools for indoor information built on geographic information systems, online analytical processing and data mining will appear.

Organizational Benefits

Large firms will become more inclined to adopt IPSs as they will provide new seamlessly integrated functionalities to their existing information system infrastructure (e. g., ERP, CRM, SCM). The convergence will give rise to novel applications that will contribute to increase organizational performance. Without a doubt, the strategic advantage of indoor positioning will decrease as its technology and applications become ubiquitous in business processes. Benefits will be most important in inventory management, merchandise tracking, collaborative work, security in organizations, asset management and supply chain optimization.

Individual Benefits

The extent of adoption of IPSs by organizations largely depends on the benefits perceived by their employees. IPSs can improve employees' work and improve well-being: context-aware information retrieval, simplified automated management of rooms access in work facilities, decision support system for public safety (e. g., children at school), life-critical reduction in time of emergency response.

People and Society

Advances in indoor positioning must be mirrored by research spanning human (e. g., privacy), ethical, legal (e. g., access to personal data) and social issues. Indeed, the control or perception of control caused by IPSs can lead to diverse reactions from uneasiness to a plain refusal to adopt the technology.

Cross References

► Statistical Descriptions of Spatial Patterns

Recommended Reading

1. Dempsey, M.: The fundamentals of indoor positioning systems in healthcare. Biomed. Instrum. Technol. 293–297 (2003), (PDF version on the web: www.radianse.com/download/newsbit-2004.pdf, July 12th, 2007)
2. Zeimpekis, V., Giaglis, G. M., Lekakos, G.: A taxonomy of indoor and outdoor positioning techniques for mobile location services. J. ACM Special Interest Group Electron. Commerce **3**, 19–27 (2003)
3. Hightower, J., Borriello, G.: Location systems for ubiquitous computing. IEEE Comput. **34**, 57–66 (2001)
4. Want, R, Hopper, A., Falcão, V., Gibbons, J.: The Active Badge Location System. ACM Trans. Inf. Syst. **10**, pp. 91–102 (1992)
5. Weiser, M.:. The computer for the 21st century. *Sci. Am.* **265**, 94–104 (1991), July 12th, 2007
6. Weiser, M.: Some computer science issues in ubiquitous computing. Commun. ACM **36**, 74–84 (1993)
7. Gow, G.A.: Privacy and ubiquitous network societies. Paper presented at the ITU workshop on ubiquitous network societies, International Telecommunication Union, April 6–8, (2005) Geneva, Switzerland
8. Beresford, A.R.: Location privacy in ubiquitous computing. In: Technical Report No. 612 UCAM-CL-TR-612, University of Cambridge, Computer Laboratory, UK (2005)

9. Srivastava, L.: Ubiquitous network societies: the case of radio frequency identification. Paper presented at: the ITU workshop on ubiquitous network societies, International Telecommunication Union, April 6–8 2005
10. Perreault L., Lapierre C., Plante M., Bachy D., Caron C.: MicroGeomatics: a new paradigm for mobile workforce management. GeoWorld **March**, 26–28 (2006)
11. O'Connor, M.C.: RFID takes attendance—and heat. RFID J. **February 16** (2005)
12. Muthukrishnan, K., Lijding, M., Havinga, P.: (2005). Towards smart surroundings: enabling techniques and technologies for localization. In: First International Workshop on Location- and Context-Awareness (LoCA 2005), Oberpfaffenhofen 2005, Faculty of Computer Science, University of Twente, Enschede, The Netherlands
13. Guvenc, I., Abdallah, C.T., Jordan, R., Dedeoglu, O.: (2003). Enhancements to RSS based indoor tracking systems using Kalman filtersIn: Global Signal Processing Expo and International Signal Processing Conference, Dallas, Texas, March 31–April 3 2003, Guvenc
14. Melnikov, H.: Open solution for location based services in WLAN environment. Master thesis, Tampere University of Technology (2004)
15. Asif, Z.: Integrating the supply chain with RFID: A technical and business analysis. Commun. Assoc. Inf. Syst. **15**, 393–427 (2005)
16. Anne, M., Crowley, J.L., Devin, V., Privat, G.: Localisation intra-bâtiment multitechnologies: RFID, Wi-Fi, vision. In: Proceedings of Ubimob 2005, Grenoble, France, 2005, Vol.120, pp. 29—35
17. Pahlavan, K., Li, X., Makela, J-P.: Indoor geolocation science and technology. IEEE Commun. Mag. **40**, 112–118 (2002)
18. Battiti, R., Brunato, M., Villani, A.:. Statistical learning for location fingerprinting in wireless LANs. In: *Technical Report DIT-02–0086*, University of Trento, Informatic and Telecommunication Department, Italy (2002)
19. Ladd, A.M., Bekris, K.E., Rudys, A., Marceau, G., Kavraki, L.E., Wallach, D.S.: (2002). Robotics-based location sensing using wireless ethernet. In: 8th ACM MOBICOM, Atlanta, Georgia, September 23-28 2002
20. Xiang, Z., Song, S., Chen, J., Wang, H., Huang, J., Gao, X.: A wireless LAN-based indoor positioning technology. IBM J. Res. Dev. **48**, 617–626 (2004), Armonk, New York, USA
21. Hightower, J., Borriello, G.: A survey and taxonomy of location systems for ubiquitous computing. In: Technical Report UW CSE 01–08–03, University of Washington, Washington (2001)
22. Kaemarungsi, K.: Design of indoor positioning systems based on location fingerprinting technique. PhD thesis, University of Pittsburgh (2005)
23. Prasithsangaree, P., Krishnamurthy P., Chrysanthis, P.: On indoor position location with wireless LANs. Paper presented at: The 13th IEEE international symposium on personal, indoor and mobile radio communications (PIMRC 2002), Lisbon, 2002.
24. Jacobsen, H.-A. Middleware for location-based services. In: Schiller, J., Voisard, A. (eds.). Location-Based Services. Morgan Kaufmann. Elsevier, pp. 83–114 (2004)
25. Coleman, K.: Convergence: GIS/Communications/Information Technologies. In: *Direction Mag.* **13/11/2004**. Available via: www.directionsmag.com (2004)
26. Batty P. 2004: Future Trends & the Spatial Industry – Part Two. Geospatial Solutions. Advanstar Communications, New York, September, pp. 32-35

Indoor Positioning, Bayesian Methods

JUE YANG, XINRONG LI
Department of Electrical Engineering,
University of North Texas, Denton, TX, USA

Synonyms

Geolocation; Localization; Location estimation; Bayesian estimation; Mobile robotics; Location tracking

Definition

Indoor positioning, generally speaking, is the technology through which the geospatial location coordinates of a number of mobile or stationary objects are determined in indoor environments. A typical indoor positioning system usually estimates the target object's location from observation data collected by a set of sensing devices or sensors. When the target object is stationary the location estimation problem is also referred to as localization problem. On the other hand, estimating the location of mobile target objects is known as target tracking.

Bayesian estimation methods are based on the Bayes' theorem, a well-known result in probability theory, which relates the conditional and marginal probability distributions of random variables. Bayesian approaches are fundamentally different from the classical approaches such as maximum-likelihood estimator and minimum-variance unbiased estimator in that the unknown parameters are assumed to be deterministic constants in classical approaches but random variables in Bayesian approaches. By assigning a probability density function (PDF) to the unknown parameters, a Bayesian optimal estimator can always be determined, such as the minimum mean square error (MMSE) estimator that is optimal on the average.

Historical Background

The concept of geolocation is not new; it has evolved over many years through successful design and application of various legacy localization and tracking systems such as sonar, radar, and the global positioning system (GPS). In recent years, location-awareness of mobile wireless devices has attracted great interest from researchers in both the academic and industrial communities [1]. For example, Federal Communications Commission (FCC) mandated that all wireless carriers need to implement wireless E911 service for mobile subscribers in phases starting from 1996; location-awareness has also become a major research topic in pervasive computing and sensor networks communities [2]. Location information for people or mobile devices has tremendous potential for many innovative applications in indoor environments such as shopping centers, museums, office buildings, hospitals, and prisons. Consider that in office buildings users may often need to print to the nearest printer from laptop computers or get directions to a particular office on palmtop computers; in prison or intensive healthcare facilities the location of various personnel need to be monitored and logged continuously for management, safety and/or surveillance purposes [3,4].

The well-known GPS technology has proven to be extremely valuable in many military and civilian applications. However, GPS can only work reliably where the GPS receiver has clear unobstructed line-of-sight view of at least four NAVSTAR satellites. Building walls and other objects in and around building environments can easily scatter and attenuate the radio signals employed in most geolocation systems, which significantly degrades their performance. Thus, in recent years many alternative geolocation technologies have been proposed and studied for complex indoor environments, including enhanced GPS, location fingerprinting, superresolution time of arrival (TOA), ultra-wideband (UWB), radio-frequency identification (RFID), inertial navigation and dead reckoning, wireless local area network (WLAN)-based localization, Kalman filters, particle filters, etc. [1,4,5,6]. Among the alternatives, Bayesian methods are well suited for indoor positioning, since the Bayesian methodology provides a unified framework to conveniently incorporate prior knowledge, integrate multimodal location sensors, and exploit historical data through a recursive tracking process.

The Bayesian estimation methods have gone through many years of development. For example, over the past decades, Kalman filters, one of the instantiations of Bayesian methodology, have been successfully employed in many navigation and tracking systems [7]. More recently, however, with the tremendous advancement in computational power other Bayesian methods such as particle filters have become increasingly popular in the research of positioning techniques [8,9,10]. In this article, a brief description of the scientific fundamentals of Bayesian methods and the key applications of Bayesian methods are presented within the context of indoor positioning applications.

Scientific Fundamentals

Principles of Bayesian Methods

General Framework Bayesian methods provide a rigorous framework for dynamic state estimation problems [7]. They are intended to probabilistically estimate the location of target objects from noisy observations [11]. In the general framework of Bayesian methods or Bayesian filters, the location of an object is represented by a vector random variable $\boldsymbol{x}$, which is often specified by the coordinates in two- or three-dimensional Cartesian space, and sometimes by even including pitch, roll, yaw, and linear and rotational velocities, depending on specific application requirements. Observations are the measurement data collected from a wide variety of sensors that are available while the type of sensors that are employed depends on the specific application scenarios. Based on the observations collected from n sensors, namely $z = \{z_1, z_2, \ldots, z_n\}$, Bayesian filters estimate the possible positions of the target, depicted by the posterior PDF of the vector random variable $\boldsymbol{x}$ conditioned upon all of the available observation data, that is, $p(x|z)$.

By applying the Bayes' theorem, it can be readily derived that [7]

$$p(x|z) = \frac{p(z|x)p(x)}{p(z)} = \frac{p(z|x)p(x)}{\int p(z|x)p(x)\ \mathrm{d}x}\,, \tag{1}$$

which describes the general framework of Bayesian filters. The distribution function $p(x)$ is usually referred to as the prior PDF since it completely characterizes the initial or prior knowledge about the unknown parameter x before any observation data has been collected. The stage at which the prior distribution is established is known as the prediction stage. On the other hand, the conditional PDF $p(z|x)$ is usually referred to as the perception model, observation model, or likelihood function, which represents the likelihood of making the observation z given that the object is at the location $\boldsymbol{x}$. Within the context of Bayesian framework given in (1), the likelihood function can be seen to utilize the measurements to refine the prior knowledge of the unknown variable $\boldsymbol{x}$ to obtain the posterior distribution of the unknown variable. Thus, the stage of deriving the posterior distribution of unknown random variable from the likelihood function and the prior distribution as in (1) is often referred to as the update stage.

Recursive Bayesian Estimation In most of the tracking applications, it is often required to continuously track the location of a target object in real-time; that is, the location estimate of the target object is continuously updated as new observation data arrive. The recursive Bayesian estimation method was developed for such scenarios following the general framework of Bayesian filtering given in (1).

More specifically, from the Bayesian perspective the purpose of tracking is to determine the posterior distribution $p(x_k|z_{1:k})$ recursively, where x_k is the state at the time k and $z_{1:k} = \{z_1, z_2, \ldots, z_k\}$ is the ensemble observation data up to the time k [9]. It is assumed that the prior distribution $p(x_0) \equiv p(x_0|z_0)$ is available, where z_0 designates the case that no observation data is available. Then, the posterior distribution can be obtained recursively according to the two stages introduced above. Suppose that the posterior distribution $p(x_{k-1}|z_{k-1})$ at the time $k-1$ is available. At the prediction stage, that is, before the observation at the time k arrives, the prior PDF of the state at the time k can be obtained via the Chapman–Kolmogorov equation [9]:

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})\,\mathrm{d}x_{k-1}\,, \quad (2)$$

where $p(x_k|x_{k-1})$ is the state transition model describing the system dynamics. Typically, the state transition of a target object is defined as a Markov process of order one, so that $p(x_k|x_{k-1}, z_{1:k-1}) = p(x_k|x_{k-1})$, which is utilized in deriving (2). In addition, in defining the perception model, it is usually assumed that $p(z_k|x_k, z_{1:k-1}) = p(z_k|x_k)$. Thus, at the update stage, the posterior PDF at the time k can be derived by updating the prior using the new observation data z_k as in (1),

$$\begin{aligned} p(x_k|z_{1:k}) &= \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \\ &= \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{\int p(z_k|x_k)p(x_k|z_{1:k-1})\,\mathrm{d}x_k}\,. \end{aligned} \quad (3)$$

Statistical Modeling

The general framework introduced only provides an abstract approach for probabilistic localization and tracking. In practice, implementation of Bayesian filters requires the specification of the prior PDF $p(x)$ in localization or $p(x_0)$ in tracking, the perception model $p(z|x)$ or $p(z_k|x_k)$, and the system transition model $p(x_k|x_{k-1})$ in tracking applications. In this section, a briefly discussion is presented on how these distribution functions can be determined in practice.

Prior Model The prior distribution of the state variable $p(x_0)$ is typically derived from the prior knowledge about the target state that is available. The Bayesian framework in (1) provides a convenient way to incorporate the prior knowledge. In contrast, in classical estimation methods it is difficult to make use of any prior knowledge. It is a fundamental rule of estimation theory that the use of prior knowledge will lead to a more accurate estimator, and any prior knowledge when modeled in the Bayesian sense will improve the Bayesian estimator [7]. Thus, the choice of the prior distribution is critical in Bayesian methods. Usually, Gaussian prior PDF is employed in theoretical analysis due to its mathematical tractability. However, in actual implementation of Bayesian methods, any types of prior distributions can be easily incorporated using the particle filtering techniques (discussed in "Particle Filters"), which is a versatile approximate nonlinear Bayesian filtering technique based on sequential Monte Carlo simulations. In the absence of any prior knowledge, noninformative prior distribution is used such as the uniform distribution and the noninformative Gaussian distribution [7].

System Transition Model The state transition model $p(x_k|x_{k-1})$ is inferred from the knowledge of system dynamics such as the maneuvering direction, velocity, and acceleration of the target object in tracking applications. In tracking applications, the state transition is usually modeled as a Markov process of order one, that is [9],

$$x_k = f_k(x_{k-1}, v_{k-1}) \quad (4)$$

where $f_k(\cdot)$ is a function of the state at the time $k-1$, x_{k-1}, and the random perturbation noise v_{k-1}. As an example, consider the following simple model for a maneuvering target. Suppose a vehicle is traveling through a sensor field at a constant velocity, perturbed only by slight speed corrections due to terrain change and other unexpected causes. The perturbation to the velocity can be simply modeled as additive white Gaussian noise, i. e., $v_{k-1} \sim N(\mathbf{0}, Q_{k-1})$, so that the state transition is defined as,

$$x_k = Ax_{k-1} + Bv_{k-1}\,, \quad (5)$$

where

$$x_k = \begin{bmatrix} r_k \\ u_k \end{bmatrix}, \quad A = \begin{bmatrix} I & \tau \cdot I \\ \mathbf{0} & I \end{bmatrix}, \quad B = \begin{bmatrix} \mathbf{0} \\ I \end{bmatrix}, \quad (6)$$

and r_k and u_k are the location coordinates and velocity of the target object at the time k, respectively, and τ is the time interval between samples, while I and 0 are the identity matrix and the all-zero matrix of appropriate dimensions, respectively. Then, the state transition model $p(x_k|x_{k-1})$ can be readily derived from the known statistical distribution of v_{k-1}.

The perturbation to the velocity can also be modeled with a random acceleration; that is, the random noise, $v_{k-1} \sim N(\mathbf{0}, Q_{k-1})$, represents random acceleration of the target object at the time $k-1$, caused by slight speed corrections. For this new modeling strategy, the state transition model remains the same as in (5) and (6) while the system matrices A and B take a slightly different form as compared with (6), i. e.,

$$A = \begin{bmatrix} I & \tau \cdot I \\ \mathbf{0} & I \end{bmatrix}, \quad B = \begin{bmatrix} \tau^2/2 \cdot I \\ \tau \cdot I \end{bmatrix}. \quad (7)$$

Both models have been used extensively in the literature for tracking applications.

Perception Model The well-known GPS technology has proven to be extremely valuable in many practical applications. However, GPS cannot work reliably in indoor environments due to severe attenuation and scattering effects caused by building walls and other objects in and around building environments [1]. As a result, in recent years many alternative geolocation technologies have been proposed and studied for indoor environments. Different types of location sensors employ different types of sensing modalities, including but not limited to the proximity, TOA, time difference of arrival (TDOA), angle of arrival (AOA), and received signal strength (RSS) of radio frequency (RF) (either narrowband, wideband, or UWB), infrared, and/or acoustic signals [1,4,5,6]. In addition, due to the complex nature of the indoor environments, extensive measurement and modeling efforts (sometimes known as training) are needed before system deployment to derive accurate perception model of each one of the location sensors by accounting for the complex effects of environmental characteristics.

In tracking applications, it is normally assumed that the observation data from location sensors z_k, contaminated by sensor observation noise n_k, only depend on the target's current location coordinates, i. e., the state variable x_k; that is,

$$z_k = h_k(x_k, n_k) \tag{8}$$

where $h_k(\cdot)$ is usually a nonlinear function of the state x_k and the observation noise n_k. Then, the perception model $p(z_k|x_k)$ can be derived from (8). For instance, suppose an array of acoustic sensors are used to measure the received energy of an acoustic signal originated from the target object. If the object emits the signal isotropically with strength A, according to the acoustic wave propagation theory, the measured energy at the ith acoustic sensor can be determined from

$$z_{k,i} = \frac{A}{\|x_k - r_i\|^{\beta}} + n_{k,i}\,, \tag{9}$$

where r_i and $n_{k,i}$ are the location coordinates and the observation noise of the ith sensor, respectively. The parameter β describes the signal attenuation characteristics of the medium, through which the acoustic signal propagates, and it is typically determined from an extensive measurement campaign within the intended application environments. Then, the perception model $p(z_k|x_k)$ can be readily derived from (9) with a specific assumption or knowledge of the statistical distribution of the observation noise $n_{k,i}$.

Practical Implementation of Bayesian Filters

The basic framework presented in "Statistical Modeling" provides a conceptual solution to the Bayesian estimation problems. Closed-form optimal solutions do exist in a restrictive set of cases, but in general, due to the complexity of the state transition model, perception model, and/or the prior distribution in many practical applications, the optimal solution of Bayesian methods cannot be determined analytically in the closed form. In this section, several practical implementations of Bayesian filters are briefly presented.

Kalman Filters In the derivation of Kalman filters, it is assumed that the posterior density at every time step is Gaussian and, hence, parameterized by a mean and covariance [7,9,12]. Kalman filters are the optimal estimators in the Bayesian sense when the system dynamics $f_k(x_{k-1}, v_{k-1})$ and the observation model $h_k(x_k, n_k)$ are both linear functions of the state variable x_k and both v_{k-1} and n_k are zero-mean white Gaussian noises.

In some applications, however, $f_k(x_{k-1}, v_{k-1})$ and $h_k(x_k, n_k)$ are nonlinear functions. In such cases, the extended Kalman filter (EKF) can be applied, which is based on local linearization of the nonlinear functions, typically using the first-order Taylor series expansion [7]. Since EKF is based on dynamic linearization, it has no optimality properties and its performance significantly depends on the accuracy of the linearization. In addition, in the derivation of EKF the state transition and observation models are still assumed Gaussian. Therefore, the performance of EKF also significantly depends on the suitability of the Gaussian assumption; that is, if the true densities are significantly different from Gaussian distribution, such as multimodal or heavily skewed, the approximate nonlinear EKF will result in poor performance.

Besides the EKF, some numerical integration methods have also been developed to approximately implement Kalman filters for nonlinear problems, when the noises are additive and Gaussian. The most widely used approximate nonlinear Kalman filters include the Gauss–Hermite Kalman, unscented Kalman, and Monte Carlo Kalman filters [13]. In all of these approximate nonlinear methods, integrals are approximated by discrete finite sums and direct evaluation of the Jacobian matrices is avoided.

Grid-Based Methods When the state space is discrete and consists of a finite number of states $\{x_k^i, 1 \le i \le N_S\}$ at the time k, the integrals in (2) and (3) can be simplified to summation [9]. The grid-based methods are optimal when the state space is finite. When the state space is continuous, in many applications the grid-based methods can also be used as an approximation method by piecewise discretization of the state space and the distribution functions. The grid must be sufficiently dense to achieve good approximation to the continuous state space. Thus, the grid density depends directly on the performance requirements of specific applications. For example, for many indoor positioning applications, the indoor environment of interest can be typically tessellated into uniform (or nonuniform) grid cells of about 1 m in size. Then, the posterior probabilities at the center of the cell or the average probabilities within the cells are determined. The major advantage of the approximate grid-based methods lies in the fact that arbitrary distribution functions can be approximated through discretization while a disadvantage of such methods is that predefined grids may result in inadequate resolution in high probability density regions. The computational complexity and memory space cost of the grid-based methods increases exponentially with the number of state space dimensions. As a result, the approximate grid-based methods are usually applied in low-dimensional applications, such as estimating only the target's location and orientation.

In addition, many indoor environments provide a natural way to represent a person's location at a symbolic level such as the room or hallway the person is currently in. Therefore, in some applications, if only imprecise location information is required, another type of grid-based method can be employed based on nonmetric topological representation of the physical application environments. Such a method significantly reduces the computational complexity of the approximate grid-based methods.

Particle Filters Particle filters are a set of approximate Bayesian methods based on Monte Carlo simulations. Such methods provide a convenient and attractive approach to approximate the nonlinear and non-Gaussian posterior distributions. Unlike the grid-based methods, particle filters are flexible and can conveniently overcome the difficulties encountered in dealing with multivariate, nonstandard, and multimodal distributions through the use of importance sampling techniques. The particle filters are also widely known as sequential Monte Carlo filters, the condensation algorithm, bootstrap filtering, interacting particle approximations, and survival of the fittest [8].

The key idea of particle filters is to represent the posterior PDF $p(x_k|z_{1:k})$ by a set of m weighted samples $\{x_k^i, w_k^i\}_{i=1,\ldots,m}$ [8,9]. In this representation, each ${x_k}^i$ corresponds to a possible value of the state at the time k, which can be easily drawn from an importance density $q(x_k|z_{1:k})$ and ${w_k}^i$ are non-negative numerical coefficients known as importance weights that sum up to one. Thus, the posterior density can be readily approximated by:

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^{m} w_k^i \delta(x_k - x_k^i)\,, \qquad (10)$$

where

$$\begin{aligned} x_k^i &\sim q(x_k|z_{1:k})\,,\\ w_k^i &\propto \frac{p\left(x_k^i|z_{1:k}\right)}{q\left(x_k^i|z_{1:k}\right)}\,. \end{aligned} \qquad (11)$$

In tracking applications, the weights can be determined recursively through sequential importance sampling (SIS) [9], i. e.,

$$w_k^i \propto w_{k-1}^i \frac{p\left(z_k|x_k^i\right)p\left(x_k^i|x_{k-1}^i\right)}{q\left(x_k^i|x_{k-1}^i, z_k\right)}\,, \qquad (12)$$

where $q(x_k^i|x_{k-1}^i, z_k)$ is the importance density for the transition model. Thus, with a SIS algorithm, the weights and support points are recursively updated as new observation data are received sequentially.

However, as the time step k increases, the distribution of the importance weights ${w_k}^i$ becomes more and more dispersed and skewed, which is known as the degeneracy problem [8,9]. The degeneracy problem can be practically solved by introducing an additional selection step, referred to as resampling, to eliminate the particles having low importance weights and multiply particles having high importance weights. A wide variety of particles filters have been developed in recent years, such as the auxiliary and regularized particle filters. When either the transition model or the observation model is linear, the Rao–Blackwell simplifications can be applied to the particle filters to reduce computational costs. A further and detailed discussion about particle filters can be found in [8] and many references therein.

Key Applications

In recent years, Bayesian estimation methods have been successfully employed in many indoor positioning and many other related applications. In this section, a brief review of several typical applications of Bayesian filters is presented within the context of indoor positioning applications.

Perception Model-Based Localization Indoors

The Bayesian methods presented in this article are well suited for many indoor localization and tracking applications. In indoor environments, extensive measurement campaigns can be carried out before system deployment so that a wide variety of prior knowledge can be derived from the premeasurements. The Bayesian framework provides a convenient way to incorporate and exploit the prior knowledge, which constitutes a major advantage of the Bayesian methods for indoor positioning applications as compared with the classical estimation methods.

As discussed earlier, extensive measurement and modeling effort is necessary in the Bayesian methods to derive accurate prior PDF and the perception model as well as the state transition model. In this section, the focus is on Bayesian localization techniques based on homogeneous sensors; localization using decision-level sensor fusion for heterogeneous sensors is presented in the next section. Once system models are established through premeasurement data, the posterior density can be readily derived following the general framework of Bayesian methods.

In [6], a radio-frequency (RF)-based system is presented, known as RADAR, for locating and tracking users inside buildings. The basic idea underlying RADAR is based on empirical perception models of the RSS measurements of the signals received from multiple base stations. With extensive premeasurement or training data collected in the training phase, a nearest-neighbor approach is applied to determine the location estimate of the target objects. Strictly speaking, such an approach is not a probabilistic method. However, it could be regarded as a special case within the general framework of Bayes estimation methods since the premeasurement data is actually employed to develop empirical perception models of the observed RSS data. The technique presented in [14] is very similar to RADAR, but the location estimation problem is addressed in a more general setting. An improvement introduced in [14] is that multiple observation vectors are collected at each specified location to provide more robust probabilistic estimation results. In [15], a more sophisticated and detailed perception model is developed, which is based on the parametric signal propagation pathloss model. The parameters in the pathloss model are typically environment specific, and thus can only be determined from extensive measurement data through linear regression analysis.

Localization Using Decision-Level Sensor Fusion

No matter how well the perception model is refined, a positioning system that is solely based on one sensing modality can only provide a limited estimation accuracy and reliability due to hardware limitations and adverse effects of application environments. In fact, the homogeneous localization techniques presented in the previous section do not fully exploit the potential of indoor applications. Inside a building, there are usually more than one existing infrastructure networks that can implicitly provide location information. For example, the wireless LAN infrastructure can work together with camera and sound surveillance systems to improve the location estimation accuracy. A robust and scalable probabilistic positioning method is presented in [16], which employs decision-level sensor fusion, to incorporate multimodal data obtained by many different types of location sensors.

In addition, three generic architectures are presented in [12] for sensor fusion systems, which only differ in terms of at which stage the fusion occurs. Decision-level sensor fusion occurs after subsystems have processed the raw observation data collected from their sensors and reported their decisions about the target's location. Each sensor module provides the entire system with a set of possible values (object locations) represented with a probability distribution. Then, the probability distributions are combined to compute a new probability distribution that represents the most likely location of the object. More specifically, following the general framework of Bayesian methods in (1), if the observation data from n sensors are $z = \{z_1, z_2, \ldots, z_n\}$,

$$p(x|z_1, z_2, \ldots, z_n) = \frac{p(z_1, z_2, \ldots, z_n|x)p(x)}{p(z_1, z_2, \ldots, z_n)} . \tag{13}$$

Then, knowing that $\{z_1, z_2, \ldots, z_n\}$ are measurement data from statistically independent sensors, it can be easily derived that:

$$p(x|z_1, z_2, \ldots, z_n) = \frac{p(z_1|x)p(z_2|x)\ldots p(z_n|x)p(x)}{p(z_1, z_2, \ldots, z_n)} . \tag{14}$$

It should be noted that the number of measurement data n is not a constant number; instead, its value varies from time to time, depending on the number of available sensors that can provide decision for the current estimation. This approach thus facilitates modular and extensible system architecture. There is no limit on the number and the types of sensors that can be employed using this method. With decision-level sensor fusion, when some of the sensors fail (or are shut off due to power saving efforts), the quality of localization is affected, but the system as a whole will continue remain functional.

Robot Tracking Using Particle Filters

The examples presented in the previous sections do not fully exploit the full strength of the Bayesian methods. The particle filters as well as other sequential Bayesian methods are best suited for location tracking problems as in many mobile robotics applications [8].

Location tracking is the most fundamental problem in many practical mobile robotics applications, and thus it has received tremendous attention in recent years. In such problems, the initial robot's location is represented by the initial prior distribution $p(x_0)$ and the location is updated by tracking techniques based on sequentially collected sensor measurements. In order to implement the particle filters, three probability distributions need to be determined first, i. e., the prior PDF $p(x_0)$, the transition model or the motion model $p(x_k|x_{k-1}, U_{k-1})$ where U_{k-1} denotes control data or odometer data, and the perception model or the likelihood function $p(z_k|x_k)$. Then, the particle filters or other recursive Bayesian estimation methods that are presented previously in this article can be applied in a straightforward way. Detailed convergence property results of the particle filters and many improvement techniques can be found in [8].

Data Association for Multiple Target Tracking

Under the most general setup of a target tracking system, which is common in many applications, a varying number of moving targets need to be tracked continuously in a region from noisy observation data that are sampled at random intervals. Such a problem is widely known as multitarget tracking (MTT) [17,18,19,20]. In MTT, the association between the observation data and the targets are typically unknown so that MTT is much more complex than the single–target tracking problem. Existing MTT algorithms generally present two basic ingredients: an estimation algorithm and a data association method [20]. To address the data association problem in MTT, some assumptions are commonly made, for example, one measurement can originate from one target or from the clutter, one target can produce zero or one measurement at a time, and one target can produce zero or several measurements at one time. Over the years, many MTT methods have been proposed in the literature based on some or all of the preceding assumptions and they vary largely on the association methods that are employed.

The most successful multiple target tracking algorithm is the multiple hypothesis tracker (MHT) [17]. With MHT, each hypothesis associates past observations with a target and, as a new set of observation data arrives, a new set of hypotheses is formed from the previous hypotheses. Each hypothesis is scored by its posterior and the algorithm returns the hypothesis with the highest posterior as a solution. A different approach to the data association problem is the joint probabilistic data association filter (JPDAF) [18], which is a suboptimal single-stage approximation to the optimal Bayesian filter. The JPDAF assumes a fixed number of targets and is a sequential tracker in which the associations between the "known" targets and the latest observation data are made sequentially. In [19], the authors presented a Markov chain Monte Carlo data association (MCMCDA) method, which shares the same ideas of particle filtering. Like other particle filtering solution, MCMCDA requires relatively low memory consumption and outperforms other algorithms in extreme conditions where there are a large number of targets, false alarms, and missing observations. In the context of MTT, particle filters are also appealing, since as the association needs to be considered only at a given time iteration, the complexity of data association can be reduced [20].

Future Directions

From the examples discussed above, it is clear that in general the probabilistic Bayesian estimation methods have tremendous potential for indoor positioning systems. In particular, the particle filtering is a versatile approximate nonlinear Bayesian filtering technique based on sequential Monte Carlo simulations and it offers a simple yet powerful implementation of Bayesian methodology for sequential inference in nonlinear, non-Gaussian systems. However, in order to employ the Bayesian methods in complex indoor environments for localization and tracking applications, accurate system models need to be developed for a wide variety of location sensors that are suitable for indoor applications. On the other hand, due to the intractability of nonlinear, non-Gaussian problems, the performance of various methods designed for such problems are typically evaluated through computer simulations. However, the performance of a geolocation system is heavily affected by relative geometric structures formed by reference nodes and target nodes (a.k.a. geometric conditioning, characterized by geometric dilution of precision (GDOP)) as well as the severe adverse multipath and non-line-of-sight (NLOS) effects in indoor environments. Therefore, accurate models of indoor radio propagation channels and other indoor environmental characteristics need to be developed to enable convenient performance study of indoor positioning systems through simulations. In addition, more theoretical developments are necessary in both algorithm design and theoretical performance bounds study for the Bayesian methods for various specific indoor geolocation applications.

Cross References

► Indoor Positioning

Recommended Reading

1. Pahlavan, K., Li, X., Makela, J.: Indoor geolocation science and technology. IEEE Commun. Mag. **40**(2), 112–118 (2002)
2. Hightower, J., Borriello, G.: Location Systems for ubiquitous computing. Computer **34**(8), 57–66 (2001)
3. Christ, T.W., Godwin, P.A., Lavigne, R.E.: A prison guard duress alarm location system. In: Proceedings of the IEEE Interna-

tional Carnahan Conference on Security Technology, vol.1, pp. 106–116, Ottawa, Ont., Canada (1993)
4. Werb, J., Lanzl, C.: Designing a positioning system for finding things and people indoors. IEEE Spectrum **35**, 71–78 (1998)
5. Ladd, A.M., Bekris, K.E., Marceau, G., Rudys, A., Wallach, D.S., Kavraki, L.E.: Using Wireless Ethernet for Localization. IROS, Lausanne (2002)
6. Bahl, P., Padmanabhan, V.N.: RADAR: An in-building RF-based user location and tracking system. IEEE Infocom **2**, 775–784 (2000)
7. Kay, S.M.: Fundamentals of Statistical Signal Processing: Estimation Theory. Prentice Hall, Upper Saddle River, NY (1993)
8. Doucet, A., de Freitas, N., Gordon, N. (eds.): Sequential Monte Carlo in Practice. Springer-Verlag, New York, NY (2001)
9. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans. Signal Process. **50**, 174–188 (2002)
10. Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U.: Particle filters for positioning, navigation and tracking. IEEE Trans. Signal Process. **50**, 425–437 (2002)
11. Fox, D., Hightower, J., Kauz, H., Liao, L., Patterson, D.: Bayesian Techniques for Location Estimation. In: Proceedings of the Workshop on Location-Aware Computing (UBICOMP Conference), Seattle, WA, October 2003
12. Hall, D.L.: Mathematical Techniques in Multisensor Data Fusion. Artech, Boston, MA (1992)
13. Cui, N., Hong, L., Layne, J.R. A comparison of nonlinear filtering approaches with an application to ground target tracking. Signal Process. **85**, 1469–1492 (2005)
14. Roos, T., Myllymaki, P., Tirri, H., Misikangas, P., Sievanen, J.: A Probabilistic Approach to WLAN User Location Estimation. Int. J. Wirel. Inf. Netw. **9**(3), 155–164 (2002)
15. Roos, T., Myllymäki, P., Tirri, H.: A statistical modeling approach to location estimation. IEEE Trans. Mobile Comput. **1**, 59–69 (2002)
16. Bohn, J., Vogt, H.: Robust Probabilistic Positioning Based on High-Level Sensor Fusion and Map Knowledge. Tech. Rep. 421. Institute for Pervasive Computing, Distributed Systems Group, Swiss Federal Institute of Technology (ETH), Zurich (2003)
17. Reid, D.B.: An algorithm for tracking multiple targets. IEEE Trans. Automat. Control **24**, 843–854 (1979)
18. Bar-Shalom, Y., Fortmann, T.E.: Tracking and data association. Mathematics in Science and Engineering Series 179. Academic, San Diego (1988)
19. Oh, S., Russell, S., Sastry, S.: Markov chain Monte Carlo data association for general multiple-target tracking problems. In: Proceedings of the 43rd IEEE Conference on Decision and Control, Paradise Island, Bahamas (2004)
20. Hue, C., Le Cadre, J.-P., Perez, P.: Posterior Cramer-Rao bounds for multi-target tracking. IEEE Trans. Aerospace Electron. Syst. **42**(1), 37–49 (2006)
21. Ramadurai, V., Sichitiu, M.L.: Localization in wireless sensor networks: a probabilistic approach. Proc. of the 2003 International Conference on Wireless Networks (ICWN2003), vol.1, pp. 275–281, Las Vegas, NV, Jan. (2003)

Indoor Positioning System

► Indoor Positioning

Indoor Positioning with Wireless Local Area Networks (WLAN)

A. Kushki, K.N. Plataniotis,
A.N. Venetsanopoulos
Multimedia Laboratory, University of Toronto,
Toronto, ONT, Canada

Synonyms

WLAN localization; WLAN location estimation; WLAN location determination; WLAN geolocation; WLAN location identification; WLAN location discovery; Radiolocation; WLAN location sensing; Position location; Location based services

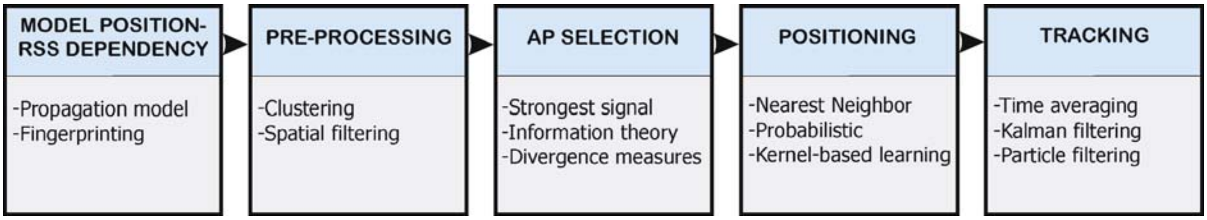
Definition

WLAN positioning refers to the process of locating mobile network devices, such as laptops or personal digital assistants, using a Wireless Local Area Network (WLAN) infrastructure. Positioning is carried out by exploiting the dependency between the location of a mobile device (MD) and characteristics of signals transmitted between the MD and a set of WLAN access points (APs). These characteristics are generally Time of Arrival (ToA), Time Difference of Arrival (TDoA), Angle of Arrival (AoA), and Received Signal Strength (RSS). RSS is the feature of choice in WLAN positioning systems as it can be obtained directly from Network Interface Cards (NIC) that are available on most handheld computers. This allows the implementation of positioning algorithms on top of existing WLAN infrastructures without the need for any additional hardware. The wide availability and ubiquitous coverage provided by WLANs makes this type of positioning a particularly cost effective solution for offering value-added location based services (LBS) in commercial and residential indoor environments.

With accuracies in the range of 1–5 meters, WLAN positioning systems can be used to infer location information in two or three dimensional Cartesian coordinates with the third dimension representing a floor number. Location information can either be reported relative to a predefined coordinate system or symbolically (e. g., room number, floor number, etc.). Positioning may be performed by the infrastructure (network-based, remote positioning) or by the MD (terminal-based, self-positioning). The former offers more computational power whereas terminal-based positioning enhances scalability and promotes privacy.

Historical Background

Cellular network infrastructures have served as a milieu for the development of positioning systems to deliver location-based emergency and commercial services, including loca-



Indoor Positioning with Wireless Local Area Networks (WLAN), Figure 1 Overview of steps involved in positioning

tion-sensitive billing and advertising, to their users since the introduction of the E-911 mandate by the U.S. Federal Communications Commission in 1996. Another important positioning system is the Global Positioning System (GPS), which offers highly accurate positioning capability and is used in outdoor navigation and LBS.

More recently, advances in wireless communication technology have led to user mobility within indoor networks, inspiring location-awareness and LBS in a wide range of personal and commercial applications geared toward indoor environments. Unfortunately, the positioning accuracy provided by existing cellular-based methods is not sufficient for such applications and coverage of the GPS system is limited in indoor environments.

With this in mind, various positioning systems have been proposed to address the problem of positioning specifically in indoor environments. These systems make use of a variety of technologies including proximity sensors, radio frequency (RF) and ultrasound badges, visual sensors, and WLAN radio signals, to carry out positioning.

Among the above methods, WLAN positioning systems have received special attention due to the fact that they can be cost-effectively implemented on top of existing and widely deployed WLANs. Similar to "cell of origin" methods in cellular systems, early work in WLAN positioning estimated the position of an MD as that of the access point with the strongest signal based on the premise that this AP is closest to the MD in the physical space. With the advent of the IEEE 802.11 based wireless networks, the first fine-grained WLAN positioning systems were introduced in the year 2000 (see, for example, the pioneering RADAR system [1]). Soon thereafter, WLAN positioning solutions were commercialized for use in applications such as asset tracking, resource management, and network security [2].

Scientific Fundamentals

The main technical challenge in WLAN positioning is the determination of the dependency between the received signal strength (RSS) and the location of an MD. As the signal travels through an ideal propagation medium (i. e., free-space), the received signal power falls off inversely proportional to the square of the distance between the receiver and transmitter. Thus, given the measurements of transmitted and received powers, the distance between the transmitter and the MD can be determined. Given three such distances, trilateration can be used to estimate the position of an MD. Unfortunately, in real environments, the propagation channel is much more complicated than the ideal scenario presented above, leading to a much more complex position-RSS dependency. The difficulties arise as a result of severe multipath and shadowing conditions as well as non-line-of-sight propagation (NLOS) caused by the presence of walls, humans, and other rigid objects. Moreover, the IEEE 802.11 WLAN operates on the license-free band of frequency 2.4 GHz which is the same as cordless phones, microwaves, BlueTooth devices, and the resonance frequency of water. This leads to time-varying interference from such devices and signal absorption by the human body, further complicating the propagation environment. To make matters worse, WLAN infrastructures are highly dynamic as access points can easily be moved or discarded, in contrast to their base-station counterparts in cellular systems which generally remain intact for long periods of time.

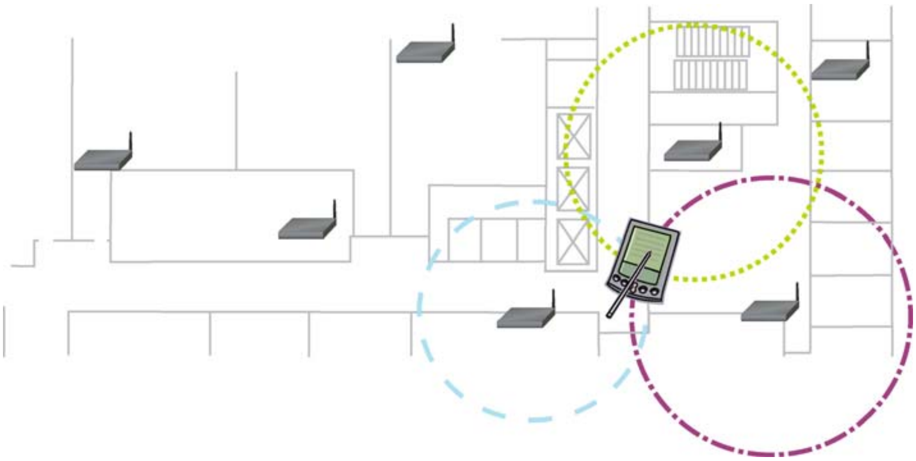
Figure 1 provides an outline of the steps involved in a positioning system. Each of the components are described in sections that follow.

Modeling of Rss-Position Dependency

Existing WLAN positioning techniques can be classified into model-based and fingerprinting approaches based on how the RSS-position dependency is determined. The approaches are discussed next.

Model-Based Methods The first class of methods aim to characterize the RSS-position dependency using theoretical models whose parameters are estimated based on training data [3]. Given an RSS measurement and this model, the distances from the MD to at least three APs are determined and trilateration is used to obtain the MD position as shown in Fig. 2.

A model for relating RSS and the distance to an AP can be constructed by considering the fact that in real envi-



Indoor Positioning with Wireless Local Area Networks (WLAN), Figure 2 Model-based WLAN positioning techniques use a propagation model to obtain distances to at least three APs and trilaterate the position of the mobile device

ronments, in addition to the distance traveled, two additional mechanisms contribute to variations in the propagation channel, namely, large scale and small scale fading. Large scale fading is due to path loss and shadowing effects. Path loss is related to dissipation of signal power over distances of 100–1000 meters. Shadowing is a result of reflection, absorption, and scattering caused by obstacles between the transmitter and receiver and occurs over distances proportional to the size of the objects [4]. Due to uncertainties in the nature and location of the blocking objects, the effects of shadowing are often characterized statistically. Specifically, a log-normal distribution is generally assumed for the ratio of transmit-to-receive power. The combined effects of path loss and shadowing can be expressed by the simplified model below [4]:

$$P_r(\mathrm{dB}) = P_t(\mathrm{dB}) + 10\log_{10} K - 10\gamma \log_{10}\left(\frac{d}{d_0}\right) - \psi(\mathrm{dB})\,. \tag{1}$$

In Eq. (1), P_r and P_t are the received and transmitted powers respectively, K is a constant relating to antenna and channel characteristics, d_0 is a reference distance for antenna far-field, and γ is the path loss exponent. Typical values of this parameter are $\gamma = 2$ for free-space and $2 \le \gamma \le 6$ for an office building with multiple floors. Finally, $\psi \sim \mathcal{N}(0, \sigma_\psi^2)$ reflects the effects of log-normal shadowing in the model.

Small scale fading is due to constructive and destructive addition from multiple signal paths (multipath) and happens over distances on the order of the carrier wavelength (for WLANs, $\lambda = \frac{c}{2.4\,\mathrm{GHz}} = 12.5\,\mathrm{cm}$). Small-scale fading effects can lead to Rayleigh or Rician distributions depending on the presence or absence of a line-of-sight, respectively [4].

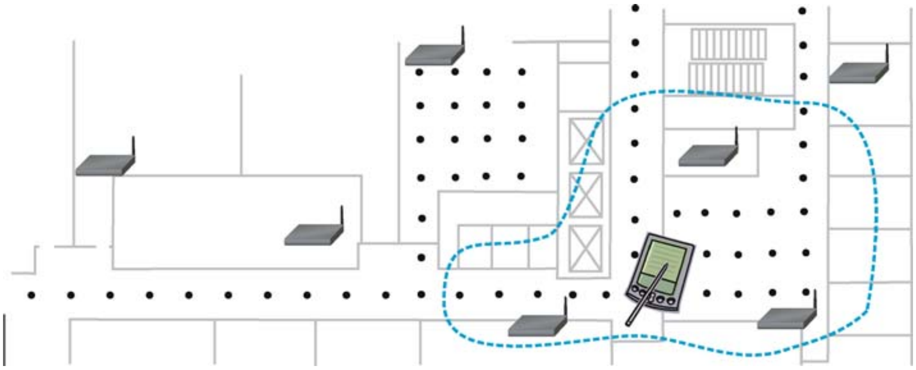
In indoor areas, materials for walls and floors, number of floors, layout of rooms, location of obstructing objects, and the size of each room have a significant effect on path loss. This makes it difficult to find a model applicable to general environments. Other limitations of model-based approaches include their dependence on prior topological information, assumption of isotropic RSS contours and invariance to receiver orientation [5].

Fingerprinting-Based Methods As an alternative to model-based methods, the RSS-position dependency can be characterized implicitly using a training-based method known as *location fingerprinting*. A location fingerprint is a vector $\mathbf{r}_i(t) = [r_i{}^1(t), \ldots r_i{}^L(t)]$ of RSS measurements from L APs at time t at spatial point $\mathbf{p}_i$. The symbol $\mathbf{p}_i = (x_i, y_i)$ or $\mathbf{p}_i = (x_i, y_i, z_i)$ represents a point in the two or three dimensional Cartesian coordinates.

Fingerprints are usually generated offline. This is done by collecting a set of measurements $\mathbf{r}_i(t)$, $t = 1, \ldots, n_i$, at a set of training locations $\{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$ with the purpose of obtaining a sufficient representation of spatio-temporal RSS properties in the given environment. The database of these location fingerprints together with their respective coordinates is known as a *radio map*. The main challenges in the construction of the radio map include the placement and number of survey points as well as determination of the number of time samples needed for a sufficient representation at each point.

Preprocessing

Before the actual positioning is performed, some preprocessing may happen to reduce computational complexity and improve efficiency of the positioning algorithm. For example, [6] proposes two preprocessing methods. The first clusters the environment into grids that receive similar AP coverage and reduces the search space to a single cluster. The second involves an incremental trilateration



Indoor Positioning with Wireless Local Area Networks (WLAN), Figure 3 Fingerprinting-based WLAN positioning techniques estimate the position of the mobile device as a combination of training points whose fingerprint record best matches the observation RSS. The dashed line delineates the training points used to form the estimate

technique where APs are used consecutively to reduce the subset of candidate locations. With the objective of power efficiency in mind, the work of [7] proposes an offline clustering method based on K-means that considers the similarity of AP values in addition to the covering sets. Lastly, [8] proposes an online spatial filtering technique to dynamically exclude irrelevant survey points from positioning calculations. The online preprocessing techniques are advantageous to their offline counterparts in terms of resiliency to loss of APs as the set of APs used during the real operation of the system may be different than that used during training.

AP Selection

Although two-dimensional positioning can be carried out with as few as three APs, a mobile device may receive coverage from many more APs in large indoor environments with ubiquitous WLAN infrastructures. Clearly, using all available APs for positioning increases the computational complexity of the system. Moreover, depending on the relative distance of the MD and each AP and the topology of the environment in terms of obstacles causing NLOS propagation, correlated RSS readings may be received from subsets of APs, leading to biased estimates.

These problems motivate the design of an AP selection block whose function is to choose the best set of APs with the given minimum cardinality for positioning. The most commonly used selection scheme is to choose APs with the highest observation RSS to ensure coverage for survey points near the observation. Unfortunately, the time variance in RSS from an AP generally increases with its mean signal strength. In such cases, the observation may differ significantly from the training values and it becomes more difficult to distinguish neighboring points.

More recently, AP selection methods employing entropy-based techniques and divergence measures have been proposed in [7,8]. In particular, the work of [7] proposes the use of the Information Gain criterion to choose APs with the best discrimination ability across the survey points and experimentally demonstrates advantages over the traditional technique of the strongest APs. Because selection is performed during the offline training of the system and remains fixed for predetermined clusters of points, the method lacks flexibility in coping with loss of APs. An online and realtime selection technique is proposed in [8] to circumvent this problem. This method aims to select a set of APs with minimum correlation to reduce redundancy. Lastly, since distance calculations are performed on the AP set chosen by the selection component, it is important to consider the interplay among the selection strategy and distance measurement when designing these components [8].

Positioning

With reference to Fig. 1, the positioning step is initiated when a new RSS measurement from an MD is received during the online operation of the system. As previously mentioned, model-based techniques rely on trilateration for positioning. In contrast, fingerprinting-based methods compare the observation to fingerprints in the radio map and return a position estimate. In general, the estimate is a combination of survey points whose fingerprints most closely match the observation.

As shown in Fig. 3, the aim of positioning is to determine a position estimate as a function of the available survey points. That is, the goal is to find $\hat{\mathbf{p}} = h(\mathbf{p}_1, \ldots, \mathbf{p}_N)$ where $\hat{\mathbf{p}}$ denotes the position estimate and $\{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$ is the set of survey points in the radio map. If $h(\cdot)$ is restricted to be a linear function of the survey points, the position estimate $\hat{\mathbf{p}}$ can be obtained as follows:

$$\hat{\mathbf{p}} = \frac{1}{K} \frac{\sum_{i=1}^{K} w_i \mathbf{p}_{(i)}}{\sum_{K=1}^{N} w_i}, \qquad (2)$$

where w_i is inversely proportional to the distance between the fingerprints at $\mathbf{p}_{(i)}$ and the observation and the set

$\{\mathbf{p}_{(1)}, \ldots, \mathbf{p}_{(K)}\}$ denotes the ordering of survey points with respect to w_i.

There are three groups of positioning techniques in the existing literature, namely, deterministic, probabilistic, and kernel-based learning methods, based on how they determine the weights w_i. These are discussed in what follows.

In the simplest case, the survey points are ranked based on the Euclidean distance between the observation and the sample mean of the RSS training samples as in RADAR [1]. In this case, the position estimate is obtained as the average of the K nearest neighbors (KNN) and $w_1 = w_2 = \ldots = w_K$.

In the more general case, the weights w_i are determined as functions of the distance between the observation RSS and the training RSS record at each survey point. For example, the weights can be chosen to be inversely proportional to the Euclidean distance above. Despite its simplicity, however, the Euclidean distance may fail to deliver adequate performance in cases where the distribution of RSS training vectors included in the fingerprints are non-convex and multimodal. Such distributions arise frequently in indoor WLAN settings due to NLOS propagation and presence of users [5].

In Bayesian approaches, such as [6], the weights are directly proportional to the likelihood or posterior probabilities $p(\mathbf{r}|\mathbf{p}_i)$ and $p(\mathbf{p}_i|\mathbf{r})$. These probabilities can be estimated from the training data either parametrically, through the assumption of a specific form for the density (e. g., a Gaussian), or nonparametrically, using density estimates such as the histogram or the kernel density estimator (KDE) [2,9]. Using the probabilistic weights, the position estimate corresponds to the maximum likelihood (ML) or maximum a posteriori (MAP) estimate when $K = 1$, and the minimum mean square error estimate (MMSE) when $K = N$ [9].

Motivated by the complexity of RSS patterns in this Euclidean space, the work of [8] proposes a kernelized distance for the calculation of the distance between an RSS observation and the fingerprints. This method nonlinearly maps the original fingerprint data to a high dimensional feature space where the distribution of RSS training vectors is simplified and carries out distance calculations in such a space. The weights are then obtained as inner products in the kernel-defined feature space. The authors show that this position estimator can be interpreted as a multidimensional kernel regression as well as the MMSE estimator of position in the case where the empirical probability density estimate (epdf) is used as the prior $f(\mathbf{p})$.

Another kernel method explored in the context of WLAN positioning is the kernel support vector machine (SVM) for both classification and regression [10], aiming to find the best set of weights for interpolating the survey points to minimize the training error while controlling the model complexity. Both of the classification and regression problems are solved independently for each dimension in the physical space. In contrast to this, the work of [11] proposes a multidimensional vector regression. In particular, a nonlinear mapping between the signal and physical spaces is built by observing that the pairwise similarity in these two spaces should be consistent. The distance between an observation and the fingerprint record is obtained as the distance between their projections onto a set of canonical vectors determined through Kernel Canonical Correlation Analysis (KCCA) and used to determine the weights in (2).

As an alternative to the use of Equation (2), decision trees are used in [7] to determine the position estimate by classifying the incoming observation as coming from one of the survey points by a series of efficient tests. Such a scheme is effective in reducing complexity and, hence, improves power efficiency on mobile devices.

An important consideration in designing training-based methods, such as the aforementioned regression techniques, is resiliency to loss of APs in WLAN infrastructures. Because APs can easily be discarded or moved, the dimensionality of the RSS vector may well be different in the training and realtime operation of the system and cannot be assumed to be fixed [8].

Tracking

Human motion is generally not random but correlated over time. Therefore, at any point in time, past position estimates can be utilized to adjust the current estimate. Such dynamic tracking solutions may simply entail a running average of previous estimates [6] or rely on more sophisticated methods such as the Kalman filter [9], Markov-model based solutions [12], and the particle filter [13].

Tracking need not be limited to the use of past position information. Predictions of future locations can also be used to proactively adjust system parameters. For example, in [9], predictions are utilized to dynamically generate subsets of the radio map for use in positioning.

Key Applications

Since its conception, WLAN positioning has been used as an enabling tool for offering location-based services in indoor environments. Three examples of specific applications are outlined below.

Network Management & Security WLAN positioning solutions can be used to offer a variety of location sensitive network services such as resource allocation, traffic management, and asset tracking. In addition, such technology can promote network security by introducing location-based authorization and authentication. Many such solutions are currently commercially available [2].

Information Delivery In order to provide mobile users with seamless and transparent access to information content, location-based personalization of information delivery is needed [14]. WLAN positioning is a particularly effective solution in such cases since the necessary infrastructure is widely available in both commercial and home settings.

Context-Awareness As location is an important piece of contextual information, WLAN positioning and tracking can offer an effective solution for context-aware applications that not only respond, but proactively anticipate user needs [15].

Future Directions

Wireless local area networks provide a cost-effective infrastructure for the implementation of positioning systems in indoor environments. As large urban cities turn into giant wireless "hotspots", it becomes essential to consider the design of inexpensive outdoor WLAN positioning methods. Such solutions must be distributed and scalable to support entire cities and power-efficient to allow implementation on a variety of mobile devices with different capabilities. Lastly, location information must be secured to prevent unauthorized usage.

Cross References

► Indoor Positioning
► Privacy Preservation of GPS Traces

Recommended Reading

1. Bahl, P., Padmanabhan, V.: RADAR: an in-building RF-based user location and tracking system. In: Proceedings of IEEE Infocom, Tel Aviv, Israel, vol. 2, pp. 775–784. IEEE, Piscataway, NJ (2000)
2. EKahau: [Online]. Available: http://www.ekahau.com/ (2006)
3. Singh, R., Macchi, L., Regazzoni, C., Plataniotis, K.: A statistical modelling based location determination method using fusion in WLAN. In: Proceedings of the International Workshop on Wireless Ad-hoc Networks, London, UK. (2005)
4. Goldsmith, A.: Wireless Communications. Cambridge University Press (2005)
5. Kaemarungsi, K., Krishnamurthy, P.: Properties of indoor received signal strength for WLAN location fingerprinting. In: Proceedings of the The First Annual Int. Conf. on Mobile and Ubiquitous Systems: Networking and Services (MOBIQUITOUS), Boston, USA, pp. 14–23. IEEE, Piscataway, NJ (2004)
6. Yousief, M.: Horus: A WLAN-based indoor location determination system, Ph.D. dissertation, University of Maryland (2004)
7. Chen, Y., Yang, Q., Yin, J., Chai, X.: Power-efficient access-point selection for indoor location estimation. IEEE Trans. on Knowledge and Data Engineering **18**(7), 877–888 (2006)
8. Kushki, A., Plataniotis, K., Venetsanopoulos, A.: Kernel-based positioning in wireless local area networks. IEEE Trans. on Mobile Computing **6**(6), 689–705 (2007)
9. Kushki, A., Plataniotis, K., Venetsanopoulos, A.N.: Location tracking in wireless local area networks with adaptive radio maps. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Toulouse, France, vol. 5, pp. 741–744. IEEE, Piscataway, NJ (2006)
10. Brunato, M., Battiti, R.: Statistical learning theory for location finngerprinting in wireless lans. Computer Networks **47**(6), 825–845 (2005)
11. Pan, J., Kwok, J., Yang, Q., Chen, Y.: Multidimensional vector regression for accurate and low-cost location estimation in pervasive computing. IEEE Trans. on Knowledge and Data Engineering **18**(9), 1181–1193 (2006)
12. Haeberlen, A., Flannery, E., Ladd, A.M., Rudys, A., Wallach, D.S., Kavraki, L.E.: Practical robust localization over large-scale 802.11 wireless networks: In: Proceedings of the Tenth ACM Int. Conf. on Mobile Computing and Networking (MOBICOM), Philadelphia, PA, pp. 70–84 (2004)
13. Evennou, F., Marx, F., Novakov, E.: Map-aided indoor mobile positioning system using particlefilter. In: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, USA, vol. 4, pp. 2490–2494. IEEE, Piscataway, NJ (2005)
14. Harroud, H., Ahmed, M., Karmouch, A.: Policy-driven personalized multimedia services for mobile users. IEEE Trans. on Mobile Computing **2**(1) 16–24 (2003)
15. Patterson, C., Muntz, R., Pancake, C.: Challenges in location-aware computing. IEEE Pervasive Computing **2**(2) 80–89 (2003)

Influence Diagrams

► Bayesian Network Integration with GIS

Influence Time

► Queries in Spatio-temporal Databases, Time Parameterized

Information Fusion

► Computing Fitness of Use of Geospatial Datasets

Information Integration

► Ontology-Based Geospatial Data Integration

Information Management Systems

► Pandemics, Detection and Management

Information Presentation, Dynamic

Zhen Wen
IBM, T. J. Watson Research Center,
Hawthorne, NY, USA

Synonyms

Context-sensitive visualization; Context-aware presentation; Automatic graphics generation

Definition

Many existing and emergent applications collect data containing geographic location. A large amount of such information from diverse sources is usually involved in investigative tasks (e. g., anti-terrorist intelligence analysis and business intelligence). Geographic information plays an important role in connecting scattered information in investigations. However, investigative tasks are highly dynamic processes where information changes can not be predicted. Therefore, traditional static geographic information presentation approaches are not sufficient for coherently presenting unanticipated information. To address this challenge, dynamic presentation approaches are proposed. Based on investigation contexts, the dynamic approaches use various metrics to dynamically measure the desirability of the information, the corresponding visualization and the visual scene updates. Then, appropriate information contents are selected and dynamically incorporated into existing visual display.

Historical Background

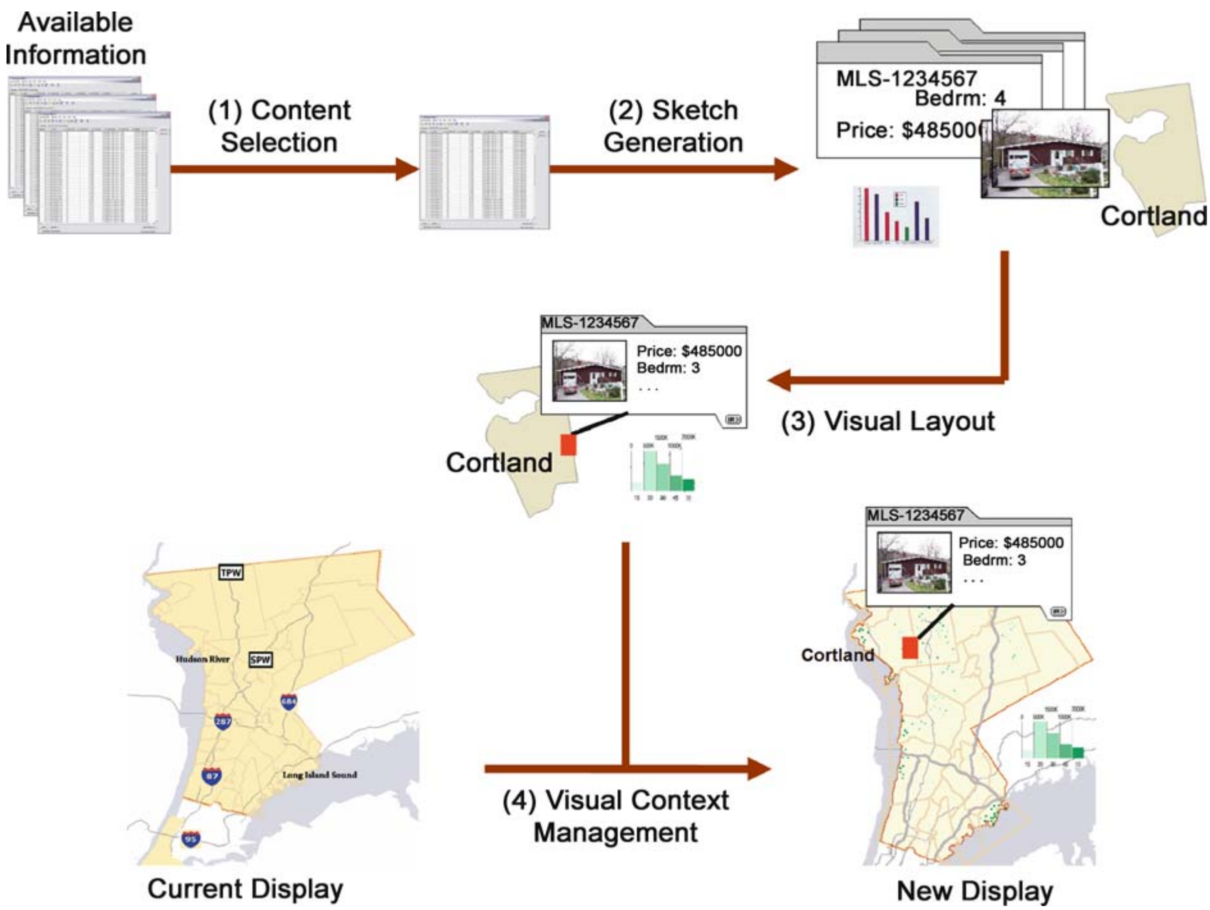
Investigative tasks, such as anti-terrorist intelligence analysis and business intelligence, require that analysts effectively distill a large amount of scattered, diverse information. Much of today's information collected by many existing and emergent applications contains geo-spatial locations. For example, most electronic transactions of daily life, such as purchasing goods by credit card or making phone calls, usually record geographic locations together with other attributes. For instance, telephone call records also include locations of communication endpoints, billing accounts, and sometimes cell phone zones. Therefore, geographic information can bridge spatially related information together into a coherent presentation which helps analysts understand and connect scattered information. For example, geographic information is essential for investigation of bio-terrorism [16] and military intelligence [9].

Traditionally, studies on geographic information presentation are mostly focused on creating maps for a pre-defined set of spatial information (e. g., cities and roads) [5]. In investigative tasks, however, analysts always need to interact with a highly dynamic set of geography related information because new information can keep coming in and may not be anticipated. For example, in a business intelligence application, analysts use geography related information to evaluate a specific real estate market. Analysts may start with an existing map to investigate the area. As the investigation unfolds, analysts may need to investigate various aspects (e. g., school districts, local business, transportation and etc.). Overlaying such related information on top of the map could assist analysts by using spatial cues to connect scattered information. Apparently it is not feasible to overlay all relevant information on a map since the information space is huge. Instead, a subset of information relevant to a current investigation focus should be presented on maps. Traditional geographic information presentation approaches [5,8] are insufficient for such purposes because three challenging issues are not addressed: (1) how to dynamically select un-anticipated information to be presented on maps; (2) how to dynamically layout unanticipated geographic information; and (3) how to transform maps as information changes to preserve an analysts' cognitive map. To address these challenges, dynamic geographic information presentation systems are demanded. Compared to traditional techniques, the dynamic presentation approaches offer two key advantages: (1) The dynamic approaches adapt the geographic information presentation as the investigation unfolds. Thus, analysts can be relieved from the burden of manually changing presentation configurations to view different spatial information during lengthy investigations; (2) The dynamic approaches ensure smooth transitions between maps containing different sets of information. Thus, analysts can better maintain their cognitive map of relevant information during investigations.

Related Work

Research in cartography [5] studied the principles of how to visualize geographic information. Cognitive scientists have also studied factors in order to facilitate information comprehension across multiple displays (e. g., maps) [12]. While these works provide empirical theories to draw upon, computational models are needed to realize these theories. In recent years, research has been initiated in the direction of supporting dynamic interaction scenarios such as those in investigative tasks.

Dynamic geographic information presentation is closely related to automated multimedia presentation systems [4,6]. These works automatically derive multimedia presentation based on user or task models. Since most of these systems support limited user interaction (e. g., [6]



Information Presentation, Dynamic, Figure 1 A pipeline for dynamic geographical information presentation

uses a predesigned menu), they often use a rule-based or schema-based approach to select content. In contrast, investigative tasks involve large and complex data sets, where developing an exhaustive set of rules or plans is impractical.

Researchers in the area of human-computer interaction and information visualization have exploited methods for dynamic overlaying information on maps. Bell et al. [3] use a greedy approach to dynamically layout labels of geographic locations during interactions which maintains a set of visual constraints, such as preventing object occlusion and ensuring visual continuity. Been et al. [1] proposed a dynamic map labeling method which is especially useful when users change map resolutions by zooming in and out. Kapler and Wright [10] presented a system called GeoTime to overlay temporal activities on maps.

Scientific Fundamentals

In a dynamic investigation, appropriate geographic information presentation can be created in four steps. First, a subset of geographic information content needs to be selected in context of the current investigation. Second, a visual sketch is created to describe the visual representations (i. e., visual encodings) of the selected information. Third, the selected information needs to be dynamically laid out on a map. Finally, a smooth transition between the previous map and the new map is derived. The four steps are illustrated in Fig. 1.

Content Selection

To provide a tailored presentation to diverse analyst queries introduced during an investigation, a geographic information presentation system must first decide the data content of the presentation. The goal of such content selection is to maximize the presentation desirability of selected information and minimize the presentation costs of the information.

To describe the presentation desirability of a set of information D, a function *desirability(D)* can be defined (for simplicity, all feature/metric values defined in this entry are normalized to lie between [0, 1]). A set of metrics can be defined to measure *desirability(D)* using the relevance of information D to the current investigative focus. The relevance can be measured in multiple dimensions.

First, certain information is more relevant in a certain application domain than other information. For example, information important to a real-estate application can be obtained by analyzing real-estate web sites (e. g., www.realtor.com and www.century21.com) in order to study the importance of various housing dimensions (e. g., price and style). In particular, the *importance* of a dimension d, $(d \in D)$ can be measured by its presentation prominence, including when it is presented (e. g., at the first click), where it is presented (e. g., at the top of a page), and how it is presented (e. g., in a bold face font). The final *importance* of a dimension is the average of its importance scores over many sites. Thus, domain relevance, *R(d, Domain)* can be defined as

$$R(d, Domain) = importance(d)\,. \tag{1}$$

Second, it is desirable to convey information relevant to analysis history, especially when current user requests are related to previous requests. For example, an analyst previously asked for the local average house price. Then, he wants to see the property tax information to understand where the heavily taxed areas are. Therefore, it is not effective to present tax information alone. Instead, previous price information should be presented together to assist the analyst to comprehend the relations. Let β denote a binary value such that $\beta = 1$ if the current query is a follow-up query and $\beta = 0$ if the current query starts a new context. Thus, history relevance can be defined as

$$R(d, History) = \beta \cdot \alpha^{t-t_0} \cdot desirability_{t-1}(d)\,, \tag{2}$$

where $0 \le \alpha \le 1$ controls how fast the previous information should decay in follow-up query scenarios; t is current time and t_0 is the time when d is first presented; $desirability_{t-1}(d)$ is the *desirability* of dimension d at previous query. By combining Eqs. 1 and 2, the desirability of a dimension d can be defined as

$$desirability(d) = \mu_1 \cdot R(d, Domain) + \mu_2 \cdot R(d, History)\,, \tag{3}$$

where μ_1 and μ_2 are weights of corresponding metrics.

Similarly, a cost function *cost(d)* can be defined to measure the presentation costs of information. One type of the presentation cost, for example, is the space cost which computes the pixels needed to convey one instance of dimension d in text or graphics. For example, the minimal space cost for displaying one house image is 100×100 pixels. Expert-made presentations (e. g., web pages) can be used to estimate the space costs required to depict a dimension (e. g., counting the minimal number of pixels needed to make a text string or an icon recognizable on a desktop).

By combining the desirability and cost functions defined above, an objective function can be define as

$$objective(D) = w_1 \cdot desirability(d) - w_2 \cdot cost(d)\,, \tag{4}$$

where $d \in D$, w_1 and w_2 are weights of the *desirability* and *cost* functions, respectively. The content selection can then be formulated as an optimization problem [13] whose objective is to maximize the desirability of all relevant metrics. A greedy algorithm [13] can be used to solve this optimization problem in real-time. Note that the metrics defined above are not intended to be a complete list of metrics. When needed, *desirability()* can also be easily extended to include other factors for dynamic presentation. For example, to adapt a presentation to different devices (e. g., PDA), a metric can also be introduced to measure how suitable the presentation is to the target device.

Sketch Generation

After the relevant information is selected, the corresponding visual encodings need to be decided. For example, use a circle to represent a city and use a information callout to show a house price. For traditional geographic information (e. g.,roads, cities, rivers), their visual encoding can be decided by existing map drawing approaches. For other types of customized information such as house information, their visual encodings can be automatically learned from an existing visualization corpus [14]. Specifically, each visualization example in the visualization corpus consists of annotations of its data and visual properties. Let tuple $\langle D_i, G_i \rangle$ denote the annotation of the i-th example in the corpus. Let D denote the data input of the dynamic presentation system. An appropriate visualization can be selected for D by

$$idx = argmin_i\{dist(D_i, D)\}\,, \tag{5}$$

where *idx* is the index of the best visualization example to select; $i = 1, \ldots, N$; and $\langle D_i, G_i \rangle$ is the i-th example in the corpus; *dist()* is a function defined to measure the difference between any two data annotations.

Visual Layout

To dynamically layout the visual encodings of the selected information, a set of visual layout constraints need to satisfied. For example, the occlusion of visual objects on a map should be avoided and spatial distribution of floating visual objects (e. g., callouts) should be balanced. In traditional static map layout approaches, the set of constraints only need to be solved once for each configuration. In contrast, in dynamic scenarios such as investigations, the set of constrained visual objects will change as the underline information changes. Therefore, the visual constants need to be dynamically maintained. To maintain such visual constraints, greedy algorithms (e. g., [2]) or non-linear constraint solvers (e. g., [7]) can be adopted.

Visual Context Management

In lengthy investigation scenarios, visual display needs to be dynamically updated to effectively incorporate new information and remove old irrelevant information. This process is called *visual context management*. It is an important process to help analysts keep their cognitive map so that they can comprehend all relevant information as a coherent whole. Specifically, let S_t denote the visual scene at time t and let S' denote the new visual information at time $t+1$. Visual context management is a process that derives a set of visual transformations (e. g., visual update operators such as *add*) which updates an existing scene S_t to incorporate new information S'.

To achieve this goal, a set of *visual momentum* metrics can be introduced to measure the desirability of derived transformations. Visual momentum measures a user's ability to extract and integrate information across multiple displays. Since the amount of visual momentum is proportional to an analysts ability to comprehend information across consecutive displays [12], the visual momentum should be maximized when a visual presentation changes. Specifically, three key techniques from [12] are most applicable to the visual context management task. They are: (1) maximizing both semantic and visual overlaps of consecutive displays, (2) preserving perceptual landmarks, and (3) ensuring smooth visual transitions.

Maximizing Both Semantic and Visual Overlaps To measure visual overlap across consecutive displays, a presentation system can compute the average invariance of each visual object in S_t and its new state in S_{t+1}:

$$O_v(S_t, S_{t+1}) = \frac{1}{N} \sum_{i}^{N} inv(v_{i,t}, v_{i,t+1}) . \tag{6}$$

Here, visual object $v_{i,t} \in S_t$, $v_{i,t+1} \in S_{t+1}$, and $v_{i,t+1} = op_i(v_{i,t})$, op_i is a visual update operator; N is the total number of visual objects in St; and *inv()* computes the invariance between two visual objects (e. g., color invariance, shape invariance).

Similarly, a semantic overlap metric can be defined to assess whether semantically related items remain together across displays. It computes the semantic relevance of S_t and S_{t+1}:

$$O_s(S_t, S_{t+1}) = \frac{1}{N^2} \sum_{i}^{N} \sum_{j}^{N} dist(d_i, d_j) , \tag{7}$$

where data objects d_i and d_j are encoded by $v_{i,t}$ and $v_{j,t+1}$, respectively, and *dist()* computes their semantic distance.

Preserving Perceptual Landmarks Perceptual landmarks are distinguishable features that anchor a visual context transition which in turn helps users to relate information in successive scenes. In a geographic information presentation system, geographic landmarks such as rivers and highways can be used to assist users to integrate information across successive displays. To preserve the maximal number of perceptual landmarks in a visual context, a metric can be defined to count the normalized number of landmarks in the context:

$$L(S_{t+1}) = \frac{L_{t+1}}{N} , \tag{8}$$

where L_{t+1} is the number of landmarks existing in visual context S_{t+1}, and N is the total number of visual objects in S_t.

Ensuring Smooth Transition Sudden changes in a scene prevents users from visually tracking the changes. As a result, the causal connection between an existing scene and a new scene may be lost. To ensure smooth transitions between successive displays, animation is often used to provide users with a powerful cue to interpret the changes. A metric can be defined to compute the average smoothness of applying a set of visual operators:

$$T(Op) = \frac{1}{K} \sum_{i}^{K} smoothness(op_i) , \tag{9}$$

where $Op = \{op_1, \ldots, op_K\}$ is the set of visual update operators used to obtain S_{t+1}. A function *smoothness()* is defined when implementing the operators to indicate how smooth the animation of the update is.

Visual Structuring Metrics In addition to maximizing visual momentum during a visual context transition, the structure of the visual scene also needs to be coherent after the transition. One example of a visual structuring issue is visual clutter. Visual clutter makes it difficult to comprehend a visual scene. Therefore, the visual clutter should be minimized after the integration of new information. Visual clutter can be measured using a set of factors:

$$\begin{aligned} C(S_{t+1}) = {} & \lambda_1 \cdot colorVariety(S_{t+1}) \\ & + \lambda_2 \cdot areaUsage(S_{t+1}) \\ & + \lambda_3 \cdot shapeComplexity(S_{t+1}) . \end{aligned} \tag{10}$$

Here, weights $\lambda_1 = \lambda_2 = \lambda_3 = 0.33$, *colorVariety()* obtains the number of different colors used in S_{t+1}. Metric *areaUsage()* computes the normalized screen space occupied by S_{t+1}. Metric *shapeComplexity()* computes the average shape complexities of the shapes in S_{t+1}. To compute *shapeComplexity()*, a complexity value is assigned to each distinct shape type to indicate the cognitive effort needed to comprehend a shape. For example, a dot in a scattered plot requires much less effect to recognize than a text label.

By combining the metrics defined in Eqs. 6, 7, 8, 9, 10, an overall objective function can be defined to measure the desirability of a scene update.

$$reward(Op, S_t, S') = u_1 \cdot (O_v + O_s) + u_2 \cdot L + u_3 \cdot T + u_4 \cdot C \quad (11)$$

where u_1, u_2, u_3 and u_4 are weights of the metrics. The arguments of each metric are omitted for conciseness. Then, the visual context management problem is formulated into an optimization problem [11]. Note that this optimization could be a non-linear optimization because the defined metrics are not linear. To solve this optimization, either a greedy algorithm or an interactive non-linear algorithm can be used. Because the involved operators and visual objects are limited in [11], a simulated annealing algorithm is used where it still achieves real-time performance.

Key Applications

Investigative tasks prevail in many application domains. Geographic information are essential for many such tasks, such as location-based business intelligence and criminal investigation. Dynamic geographic information presentation is a valuable tool for these application scenarios.

Real-Estate Information Seeking

To evaluate the house prices and their trend in certain areas, a large amount of spatial information needs to be investigated. For example, local school districts affect buying decisions for people with children. The proximities to major business, cities and transportation impact house values. Even small roads and parks influence people's perception and thus the house prices. Dynamic geographic information presentation systems help analysts to view such related information integrated on maps. Moreover, when analysts switch their current investigation focus, (e.g, from schools near a house to parks near the house), the presentation can be adapted to the changing interests. For example, less relevant information (e. g., schools) may be removed due to limited space. However, the house is still preserved to assist analysts in comprehending the information in context.

A multi-modal conversation system called RIA has been developed by IBM T. J. Watson Research Center to assist users in exploring large and complex real-estate data sets [15]. RIA has implemented the content selection and visual context management components using Java on Windows. The real-estate data is subscribed from a multiple listing service, containing 2000+ houses, each with 128 attributes (e. g., price and style). RIA supports natural language- like queries where users can request for any combinations of attributes and related concepts (e. g., schools) in context. Based on the understanding of a user request, RIA automatically creates its response in two steps. First, RIA determines the data content of the response using content selection. Second, RIA designs the form of the response using suitable media and presentation techniques so that new information can be effectively integrated with the current visual scene.

It takes two steps to set up and use the content selection or visual context management method. First, static data and visual features used in metrics need to be defined (e. g., assigning data semantic categories). Building a simple data ontology helps to define these features. Second, feature-based metrics need to be formulated to model various constraints important to an application. For example, in a mobile application, device-dependent visual context management constraints can be modeled. To bootstrap the process and avoid tuning fallacy, it is recommended to start with simple settings. Thus far, RIA has used a simple data ontology, a set of basic visual operators, and equally weighted metrics to adequately handle diverse interaction situations in the real-estate applications.

Location-based Business Intelligence

In many business, location is a very important factor. Therefore, dynamic geographic information presentation could effectively help analysts evaluate this factor in lengthy business investigations. Here, examples are given on store location planning.

When a company wants to open a new store, it also needs to investigate many factors related to spatial information. For example, for a fast food company, it needs to understand the local traffic pattern, local business and attractions, and people living nearby. Again, dynamic geographic information presentation systems can integrate such related information on maps to help analysts comprehend it as a whole. Meanwhile, the presentations are tailored to analysts' current focus (e. g., traffic) to help them attend to their current information needs.

Anti-terrorist and Criminal Investigation

For anti-terrorist and criminal investigation, where an event transpires is of great importance. Many existing tools try to incorporate maps as a component. However, other related information may often need to be viewed separately from the map. More over, a general purpose map may often contain information irrelevant to the current investigation. The irrelevant information can potentially be a distraction to analysts. Dynamic presentation systems could solve these two problems by integrating relevant information dynamically on a map based on current investigation contexts.

Public Health

To effectively prevent an epidemic outbreak, reported cases need to be tracked and analyzed in real-time. As new cases occur in different places, dynamic presentation systems can dynamically integrate customized case information on maps to assist such analysis.

Personal Web Research

With more and more information available on the Web, more people are using the Web as a personal research tool for tasks such as trip planning and relocation. In this type of research, users usually need to go through various information sources to find and evaluate related information. For example, for planning a trip, a user may go through airline, hotel, and local government information as well as reviews and opinions. Dynamic presentation systems provide a visual tool to connect scattered information spatially on-demand.

Future Directions

Software tools for investigation are receiving more and more attention because of increasingly overwhelming information. Geographic information is important to many investigative tasks that involve location related information. Dynamic geographic information presentation approaches are promising for investigative scenarios because investigations are highly dynamic.

Even though the dynamic information presentation concept has been studied for over 10 years, there are still many aspects to be improved upon in the future. For example, in many scenarios, such as text documents, information is unstructured. How to reliably assess the information relevance in dynamic investigations is still an ongoing problem.

Cross References

► Visual Momentum

Recommended Reading

1. Been, K., Daiches, E., Yap, C.: Dynamic map labeling. In: Proc. of IEEE Information Visualization (2006)
2. Bell, B., Feiner, S.: Dynamic space management for user interfaces. In: Proc. UIST ’00, pp. 239–248 (2000)
3. Bell, B., Feiner, S., Hoellerer, T.: View management for virtual and augmented reality. In: Proc. of UIST, pp. 101–110 (2001)
4. Bordegoni, M., Faconti, G., Feiner, S., Maybury, M., Rist, T., Ruggieri, R., Trahanias, P., Wilson, M.: A standard reference model for intelligent multimedia presentation systems. Comp. Stand. Interf. **18**(6–7), 477–496 (1997)
5. Brewer, C.: Designing Better Maps. ESRI Press (2005)
6. Feiner, S., McKeown, K.: Automating the generation of coordinated multimedia. IEEE Computer **24**(10), 33–41 (1994)
7. Hosobe, H.: A modular geometric constraint solver for user interface applications. In: Proc. UIST ’01, pp. 91–100. ACM (2001)
8. Imhoff, E.: Positioning names on maps. The American Cartographer **2**(2), 128–144 (1975)
9. Jones, P.M., et al.: Coraven: modeling and design of a multimedia intelligent infrastructure for collaborative intelligence analysis. IEEE Trans. Syst., Man, Cybernet. **1**, 914–919 (1998)
10. Kapler, T., Wright, W.: Geotime information visualization. In: InfoVis’04, pp. 25–32 (2004)
11. Wen, Z., Zhou, M., Aggarwal, V.: An optimization-based approach to dynamic visual context management. In: InfoVis’05, pp. 111–118 (2005)
12. Woods, D.D.: Visual momentum: A concept to improve the cognitive coupling of person and computer. Int. J. Man-Machine Stud. **21**, 229–44 (1984)
13. Zhou, M., Aggarwal, V.: An optimization-based approach to dynamic data content selection in intelligent multimedia interfaces. In: Proc. UIST ’04, pp. 227–236 (2004)
14. Zhou, M., Chen, M.: Automated generation of graphic sketches by examples. In: IJCAI ’03, pp. 65–71 (2003)
15. Zhou, M., et al.: Enabling context-sensitive information seeking. In: Intelligent User Interfaces, pp. 116–123 (2006)
16. Zubieta, J.C., Skinner, R., Dean, A.G.: Initiating informatics and gis support for a field investigation of bioterrorism: The new jersey anthrax experience. Int. J. Health Geograph. **2**(8). http://www.ij-healthgeographics.com/content/2/1/8 (2003)

Information Retrieval

► Internet-Based Spatial Information Retrieval
► Retrieval Algorithms, Spatial
► Skyline Queries

Information Services, Geography

BIN LI
Department of Geography, Central Michigan University, Mount Pleasant, MI, USA

Synonyms

GIService; GIS; Geographic information systems; Geographic information sciences; Services; Mash-ups; Location-based services

Definition

Geographic information is knowledge acquired through processing geographically referenced data. Geographic information services are (1) functionality provided by a software entity through its interfaces defined as named sets of operations and, (2) provisions of information generated from geospatial data. The development of geographic information services is closely related to distributed object technology and the Internet. It represents a major step to enable easy access to geographic information and geo-processing technology.

From software development perspectives, geographic information services represent a vertical domain of information services where many IT service components are combined to form information applications. GI-Service components are self-contained, self-describing, and reusable software objects that can be published, located, and invoked in a multiple address spaces and distributed environment. These components can be small or large and operate in different frameworks such as RPC, CORBA, DCOM, .NET, and Web Services.

Service Architecture

Different service infrastructures support different GI-Service architectures. On the desktop, DCOM/COM provides the framework for fine-grain GI-Services. On the Internet, DCOM/.NET, CORBA, J2EE, and Web Services support the distribution of GI-Services. Regardless the underlying framework, the logical architecture for distributed GI-Services has four tiers: human interaction services, user processing services, shared processing services, model/information management services (ISO19119). Different physical systems can be developed through different combinations of these tiers. For example, a thin client architecture would map human interaction services to the Web Browser, user processing and shared processing services to the Web Server and Application Server, model/information management services to the Data Server, resulting in a three-tier system. Alternatively, a thick client system can be developed by merging human interaction and user processing services into a stand-alone client, mapping shared processing to the Application Server. Google Earth is an example of a thick client.

Categories Based on the Open System Environment model of information services, the ISO19119 groups geographic information services into the following six categories:

1. Geographic human interaction services—presentation of information. Examples include Catalog viewer, Geographic viewer, and Geographic feature editor.
2. Geographic model/information management services—management of development, manipulation, and storage of metadata, conceptual schema, and data sets. Examples include WFS (Web Feature Services), WMS (Web Map Services), WCS (Web Coverage Services), and Catalog services.
3. Geographic workflow/task management services—provision of chain definition service and workflow enactment service.
4. Geographic processing services—provision of a variety of GIS functions including spatial, temporal, thematic, and metadata services.
5. Geographic communication services—typical examples are encoding (Geographic Markup Language, Keyhole Markup Language, ArcXML, and GeoTIFF) and compression of geographic data (JPEG2000, for example).
6. Geographic system management services—Examples include access control and usage rights management.

Components carrying out these services constitute the general architectural elements of geographic information services. Based on the service model, the Open Geospatial Consortium (OGC) and ISO have been working on a series of specifications/standards to guide the development of geographic information services.

OGC and Selective ISO Standards OGC and the ISO/TC211 are influential entities that develop specifications or standards of GI-Services. OGC is also a member of the W3C.

Encoding

1. Simple Features—Geometric object model of basic geospatial entities.
2. GML—ISO/CD 19136. Geographic Markup Language, as an XML-based standard format for network transfer of geospatial data.
3. Filter Encoding—XML expressions of queries. Styled Layer Descriptor—specification on symbolization of rendering of geospatial information.
4. Geographic Information Metadata—ISO19115 and ISO19139. Description of geospatial data collections and services.

Common Applications

1. Web Map Service—Display of maps in image formats.
2. Web Feature Service—Retrieval and update of geospatial data in GML format.
3. Web Coverage Service—Access to coverage data.
4. Web Map Context—Persistent representation of the state of a Web Map session for sharing and reuse.

5. Web Service Common—Common interfaces for all OGC Web Services.

Common Services

1. Catalogue Service—Publication and search of metadata.
2. Registry Service—Extension of the catalogue service using ebRIM.
3. Coordinate Transformation—Projection and transformation of geographic data.
4. Simple Feature Access—Access to geospatial data in different environments: SQL, CORBA, and OLE/COM.

Historical Background

The development of Geographic Information Services follows the path of mainstream information technology. In the early days (80s to mid-90s), like most computer software technology, Geographic Information System was a self-contained monolithic entity. Embedding GIS functionality in other software system or including external software into a GIS was a tremendous undertaking, which actually led to a specialty called system integration. Advances in distributed object technology in the mid to late 90s changed the landscape of software engineering. We first saw that object-orientation had become the dominate mode of programming and software design. Then came Microsoft's Object Linking and Embedding (OLE), the first OS level support for systematic interaction among software objects. With OLE, a Word document can contain an Excel document. As the scope of object services expands, OLE evolved into Component Object Model (COM), then Distributed Component Object Model (DCOM), and lately ".NET" that allows software units to interact through the Web. In parallel, the Object Management Group (OMG), an industrial consortium, developed the Common Object Request Broker Architecture (CORBA), which provides the reference model for the development of component technology such as JavaBeans, and other distributed object middlewares. Lastly, a lightweight framework for object services on the Web, "Web Services", has gained wide acceptance. These object standards specify how software units should be constructed, how they should be communicated with each other, and what types of common services the infrastructure must provide in order to assemble and execute component or object-based applications.

The GIS software industry and the academic community began to work with the general frameworks of distributed object technology. The OGC has been instrumental in formulating the reference specifications for developing service-oriented geo-spatial technology. Earlier efforts were in enabling interoperability among geo-spatial processing software programs. A reference model was established in 1996, followed by a series of abstract specifications for accessing spatial data. The OGC has since diligently followed the IT mainstream and guided the development of the GIS software industry with a series of service oriented specifications.

GIS software vendors have been quick in adapting to the object / component oriented framework, which can be illustrated with a brief product timeline from ESRI, the leader in the GIS software industry. ESRI released MapObjects, a collection of ActiveX components that can be assembled into application programs in 1996. In the next two years, the MapObjects Internet Map Server and the ArcView Internet Map Server both entered the market. In the subsequent year, the company released ArcInfo 8, marked a fundamental change with component technology with a comprehensive object model ArcObjects. The year 2000 represents a major push into "societal GIS" with the establishment of the geography network, a metadata service platform and major release of ArcIMS supporting multi-tier architecture and a workflow from map authoring to publication. In 2002, ESRI started "ArcWeb Services" to provide on-demand functionality and content with the Web Services framework. In 2006, along with a revolutionary wave of Web map services initiated by Google, Yahoo, and Microsoft, ESRI released ArcGIS 9.2 with a marketing as well as technical emphasis on service oriented architecture. ArcGIS Explorer, a thick client similar to Google Earth with more specialized GIS functions, maintains the relevance of the GIS software industry.

Scientific Fundamentals

Geographic information services are related to several layers of basic science. The first fundamental has to do with a general software model of Geographic Information Services. In the core of such a model is the classification of services, followed by the formal description of each service. Detail functions are further specified by the interfaces. Currently, object oriented design with Universal Model Language provides the common tools for developing software object models.

Because a key requirement in providing information services is message exchange, communication protocols are needed for describing service requirements and data contents. XML has showed wide acceptance as the language for development such protocols.

Ultimately, issues with service modeling, classification, and description, as well as service interoperations, will have to do with the ontology of geospatial information.

Ontology has been an active research area in Geographic Information Science. Most of the efforts, however, have not been directly affecting the development Geographic Information Services.

In the technical domain, architectural design and algorithm development are two key issues. Service oriented architecture is still under development. SOA is required to enable the consumption, composition, description, discovery, and publication of services, in a secure and reliable way. Algorithms must be redesigned to meet the efficiency expectations as there are potentially a large number of requests from the Web. Parallelism, once an active research area in GIS, is a promising approach to designing fast algorithms for geo-processing.

In the social science domain, there are several issues related to Geographic Information Services. A feasible business model is a prerequisite for providing sustainable Geographic Information Services. Establishing the legal and technical frameworks for handling liabilities is another infrastructural element. And finally, there are issues with usage rights of data and the protection of intellectual properties.

Key Applications

Location-Based Services

Location-based services are a unique category of GI-Services. They are distributed on wireless networks and often require real time processing. Client devices are often mobile phones and PDAs that have limited capabilities in processing, storage, and display. The types of services often include the following (OGC, 2005):

1. Directory service—Access online directory to find the nearest or specific place, product, or services.
2. Gateway service—Obtain position data from mobile devices through GMLC (Gateway Mobile Location Center) or MPC (Mobile Positioning Center).
3. Location utility service—Geocoding and reverse geocoding.
4. Presentation service—Graphic display of geo-information on mobile terminals.
5. Route service—Calculate the best route based on user specification.

Due to the specific characteristics of wireless networks, a special data encoding schema is developed. The ADT (Abstract Data Type) defines well-known data types and structure for location information through XML for Location Services (XLS).

Mobile Services

The number of mobile services of location information is rapidly rising. Below is a sample.

1. ESRI ArcPAD, ArcGIS Engine, ArcWeb Services
2. Google Mobile
3. MapInfo MapX Mobile
4. MapQuest Find Me (co-brand with Nextel)
5. Microsoft Windows Live Search for Mobiles
6. Microsoft MapPoint Location Service
7. Microsoft Virtual Earth for Mobiles
8. Tom Tom (navigation devices and services)
9. Webraska (GPS navigation software, SmartZone Geospatial Platform)

Desktop Applications

GI-Service components for desktop applications are often DCOM or JavaBean components. Each component exposes a set of interfaces as service contracts. Different components are glued together to form a desktop GIS application. ArcMap, a Windows GIS application, is built with ArcObjects, the core component library from ESRI. Other desktop applications, such as PowerPoint and MS Word, can also embed GIS components in their documents. Distributed object technology facilitated the transformation of GIS software systems from monolithic packages to a system of service components.

Network Applications

GI-Service components for Internet applications are often coarse grain, or large entities formed by smaller components such as those in ArcObjects. They are usually high level services combining a series of GIS functions and specific data. ESRI ArcWeb Services, for example, include five application services: mapping, place finder, address finder, route finder, and spatial query. A subscriber can use one or more of these services through a client program (ArcMap or ArcGIS Explorer, for example) or develop an application that combines the GI-services with other third party services.

Google, Yahoo, and Microsoft provide similar GI-services through the Internet. Google Maps, Yahoo Maps, and Microsoft Virtual Earth are all stand-alone Web applications. Meanwhile, each of them exposes a series of APIs that allow the "mash-up" of information from other Web applications. There are numerous of mash-ups developed everyday.

Software Products for Application

Access Control and Rights Management

CubeSERV Identity Management Server (CubeWerx)

Authoring

ArcIMS (ESRI)

FreeStyler (Galdos)

Application Servers

ArcGIS Server (ESRI)
ArcGIS Image Server (ESRI)
ArcIMS (Multiple protocol support, ESRI)
Catalina (WFS server, Galdos)
CubeSERV (WFS server, CubeWerx)
CubeSERV Cascading Server (WMS server, CubeWerx)
GeoServer (WFSserver, open source, Refraction Research)
OpenLayers (Map server, OSGeo)
RedSpider Web (WFS, WMS server, Ionic Software)
MapXtreme (MapInfo proprietary system)
MapGuide (Autodesk, WFS, WMS server)
MapServer (University of Minnesota, open source)
Catalog Servers
INdicio Catalog Server (Galdos)
CubeSERV Web Registry Server (CubeWerx)
ArcIMS Metadata Services (ESRI)
RedSpider Catalog (Ionic Software)
MapInfo Discovery (MapInfo, limited functions)
Clients
ArcCatalog (ESRI, metadata publishing and search)
ArcExplorer (ESRI)
ArcGIS Explorer (ESRI)
CubeXLOR (WMS client from CubeWerx)
Google Earth (Google)
Google Maps (Google)
Live Search (Microsoft)
MapBuilder (OSGeo)
NASA World Wind (NASA)
Yahoo Maps (Yahoo!)

Future Directions

The development of Geographic Information Services signifies a transition from monolithic software systems to interoperable components, from an emphasis in providing software tools to a focus on distributing information products, from a highly specialized technology mastered by the few to an information commodity consumed by the public, and from an information tool to a media of mass communication. The overwhelming popularity of online mapping services from Google, Yahoo, and Microsoft confirms that Geographic Information Services is becoming a ubiquitous information commodity.

With these easily accessible mapping services, people suddenly discovered their inherent interests in geography, locations, and maps, which are beyond finding places and getting directions. A piece of news is now attached with the geographic coordinates. Photos taken on a trip are linked to markers on the Web Map to be shared with friends and relatives. On these online maps, strangers meet, stories are shared, and communities are formed. Web maps are becoming the sand-box of the information society.

Hardware and software infrastructures will continue to improve, which will enhance the processing capability and quality of Geographic Information Services. The maturity of Service Oriented Architecture, for instance, will enable greater efficiency in authoring, chaining, publishing, and discovery of geographic information services. As a result, business and government entities will be able to integrate geographic information with other information in decision making more closely than ever before.

The availability of high quality data is likely to be a crucial condition for the development of Geographic Information Services. The issue is perhaps more in the institutional domain than in the technical one. Accessibility to high quality data inevitably contradict with privacy protection and even national security. Management of usage rights with geographic data is a challenge. The legal framework is still under development.

In the domain of academic research, development in analytical methods will play an important role in improving Geographic Information Services. Basic research in understanding earth processes and representing such knowledge as computational methods is the ultimate source of robust services. A better understanding in the spatial cognitive process, on the other hand, is increasingly important for effective interface design and information communication.

Cross References

► Catalogue Information Model
► Cyberinfrastructure for Spatial Data Integration
► Distributed Geospatial Computing (DGC)
► Geocollaboration
► Geography Markup Language (GML)
► Geospatial Authorizations, Efficient Enforcement
► Geospatial Semantic Integration
► Geospatial Semantic Web
► Grid, Geospatial
► Internet GIS
► Internet-Based Spatial Information Retrieval
► Location Intelligence
► Location-Based Services: Practices and Products
► Metadata and Interoperability, Geospatial
► OGC's Open Standards for Geospatial Interoperability
► Security Models, Geospatial
► Web Feature Service (WFS)
► Web Mapping and Web Cartography
► Web Services, Geospatial

Recommended Reading

1. OGC, OpenGIS Location Services (OpenLS): Core Services, OGC 05-016. http://portal.opengeospatial.org/files/?artifact_id=8836 (2005)
2. ISO 19119, Geographic Information—Services (2005)

3. Berre, A.J.: Geospatial Services. ISO/TC211 WG4 No. 43 (1996)
4. Buttenfield, B.P.: Looking Forward: Geographic Information Services and Libraries in the Future. Cartography and Geographic Information Systems, **25**(3), 161–173
5. Glass, G.: Web Services: Building Blocks for Distributed Systems, Prentice Hall, Upper Saddle River, NJ (2002)
6. Gunther, O., Muller, R.: From GISystems to GIServices: Spatial Computing on the Internet Marketplace. Proceedings of the International Conference on Interoperating Geographic Information Systems, Santa Barbara, CA, December 3–4 1997
7. Li, B.: Client-side Processing for Geographic Information Services. In: Geoinformatics for Global Change Studies and Sustainable Development, Manchu Li, et al. (eds.), Berkeley: CPGIS (2002)
8. Li, B.: A Component Perspective on Geographic Information Services, Cartography and Geographic Information Systems. **27**(1), 75–86 (2000)
9. Li, B., Zhang, L.: Distributed Spatial Catalog Service on the CORBA Object Bus. Geoinformatica. **4**(3), 253–269
10. OASIS, Reference Model for Service Oriented Architecture, version 1.0. http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf (2006)
11. OMG (Object Management Group), CORBATM/IIOPTM Specification 3.0.3. http://www.omg.org/technology/documents/formal/corba_iiop.htm
12. Peng, Z.R., Tsou, M.H.: Internet GIS: Distributed Geographic Information Services for the Internet and Wireless Network. John Wiley & Sons, Inc., New Jersey (2003)
13. Tao, C.: Online GIServices. Journal of Geospatial Engineering. **3**(2), 135–143 (2001)
14. Tsou, M.H., Buttenfield, B.P.: A Dynamic Architecture for Distributed Geographic Information Services. Transactions in GIS. **6**(4), 355–381 (2002)
15. Tsou, M.H., Buttenfield, B.P.: An Agent-based, Global User Interface for Distributed Geographic Information Services. Proceedings 7th International Symposium on Spatial Data Handling, Vancouver, British Columbia, July, 1998. 603–612 (1998)
16. Tsou, M.H.: An Operational Metadata Framework for Searching, Indexing, and Retrieving Distributed Geographic Information Services on the Internet. In Geographic Information Science (GIScience 2002), Egenhofer, M., Mark, D. (eds.), Lecture Notes in Computer Science vol 2478, pp. 313–332. Springer-Verlag, Berlin (2002)
17. Zhuang, V., Wrazien, D., Wang, M.H., Huang, X.W.: Programming ASP.NET for ArcGIS Server, Thomson Delmar Learing Inc., Clifton Park, NY (2005)
18. W3C (World Wide Web Consortium), SOAP Version 1.2, http://www.w3.org/TR/soap

Information Theory

► Data Compression for Network GIS

Information Visualization

► Exploratory Visualization

Inservice

► Intergraph: Real Time Operational Geospatial Applications

Integration

► Smallworld Software Suite

Intelligence, Geospatial

DWIGHT LANIER
Institute for Environmental & Spatial Analysis, Gainesville State College, Gainesville, GA, USA

Synonyms

GEOINT; Geo-intelligence

Definition

Geospatial Intelligence (GEOINT) is defined as the exploitation and analysis of imagery and geospatial information to describe, assess, and visually depict physical features and geographically referenced activities on Earth [1]. As a specialized field within the much larger profession of intelligence, GEOINT provides action-oriented intelligence of a unique and powerful nature. By combining imagery, imagery intelligence (IMINT), and geospatial information with data collected by various other Intelligence Community (IC) components, or INTs (e. g., human intelligence (HUMINT), signal intelligence (SIGINT), etc.), a Common Operational Picture (COP) of an area of interest is developed. This approach provides context and clarity to decision makers such as war fighters, homeland security personnel, and national security policymakers [2]. The inherent spatiotemporal reference frame or geography of these data acts as the common link that fuses together what would be individual pieces of information into multi-dimensional visualizations that allow analysts to extract higher levels of understanding about the situation at hand. The planning, collection, processing, analysis, exploitation, and dissemination of information from these many sources is integrated at multiple spatial as well as temporal scales (e. g., from vast desert areas to dense urban centers, and from eras to seconds), and leveraged to make important decisions [3]. The versatility and capability of geographic information systems to effectively handle this approach to multi-source data integration and analysis is what allows GEOINT to provide a more powerful solution to intelligence professionals than would be possible if each piece of information were examined individually.

Historical Background

The fundamental history of geospatial intelligence can be traced as far back in time as any group has set out to perform reconnaissance and mapping for the purpose of fusing that information into an intelligence cycle that drives decision making (i. e., the exploration of the Louisiana Purchase by Lewis and Clarke). Geospatial intelligence, as defined here, has come a long way from the days of hand generated maps, but has always been a truly invaluable component of the intelligence cycle.

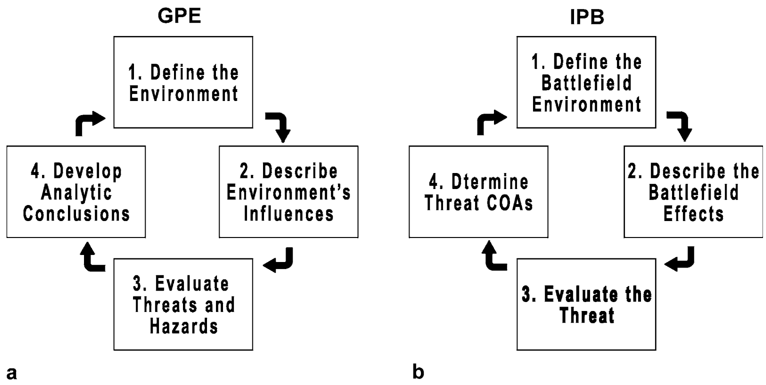
Commanders and leaders of all levels rely heavily upon knowledge of the geography of battlefields and of both foreign and domestic territories when considering their options and formulating their plans. With the advancement of the geospatial, imaging, and computer sciences (aka information technology) a more abundant source of information and improved analysis capability has found renewed strength in the decision making process. As aerial photography and photographic interpretation advanced, a wealth of information and intelligence was available about troop movements, enemy positions, and other issues and events key to national and global security. With the launch of satellites, a new era of intelligence gathering from space was ushered in. The advances of digital over film based transfer of images from satellite remote sensing platforms served to further shorten the response time to events. The tremendous boost in speed and processing capability of computers in recent times has allowed processing and analysis of geographic information (which, in its infancy was quite limited) to reach levels that finally began to unlock the potential of geographic information science. Maps that were once drawn by hand over a period of days can now be generated in a fraction of the time, and photographs that were once delayed by having to go through the delivery and exposure process now stream down as digital images to multiple users around the world in seconds. It is these events and developments, largely moved along by military necessity, that have molded what is now referred to as geospatial intelligence.

Scientific Fundamentals

Geospatial intelligence can be explored in further detail by examining the data it collects, what it does with that data on the processing side, and how the resulting products impact the intelligence community. To begin, an important distinction about the role of GEOINT in the intelligence cycle must be made. While other subject areas of the intelligence community have the ability to use GEOINT to develop their strategies and to find answers to intelligence problems, GEOINT also uses other forms of intelligence to ensure that the products it produces best match up with the overall picture. While there appears to be an overlap in the operations of GEOINT and other intelligence disciplines as far as intelligence fusion goes, it should be noted that the nature of their services is quite different. While GEOINT may bring together different components of multi-source data for the purpose of strengthening its geographic understanding and analysis of a situation, it is not to be confused with other intelligence disciplines whose sole function is to combine all sources of intelligence, including GEOINT products, and analyze them to solve intelligence problems [2].

One of the main strengths of GEOINT is that it is capable of exploiting the spatial and the temporal aspects of many sources of intelligence data. The three primary sources of data, as listed in the formal definition of GEOINT, are imagery, imagery intelligence, and geospatial information. These three components serve as the base upon which other products and data are extracted from or added to. The imagery component comes from national reconnaissance and intelligence platforms, such as commercial satellite, aerial, and unmanned aerial vehicles (UAVs). Types of imagery exploited by GEOINT cover the entire range of available remote sensing capabilities to include (but not to be limited to): panchromatic (visible), infrared, multi- and hyper- spectral, thermal, short and long wave (e. g., microwave), earth observation imagery, and other aerial platforms. Imagery collection is organized at both the local and national level with both managements typically being responsible for populating a national database [2]. Active sensor based information includes but is not limited to: Synthetic Aperture Radar (SAR), Light Detecting and Ranging (LIDAR), etc. Once imagery and other sensor information is collected and received, it is analyzed to extract valuable intelligence. The intelligence extraction step is performed by image interpreters through spectral and object-based classification techniques as well as varying forms of automated and manual feature extraction methods. The last data component, geographic information, is often a product of the first two steps. It does, however, also include other forms of data collected from surveying and mapping technologies, geodetic data, and other related products [1].

With the geographic data and imagery in hand, analysis is performed in a way that ensures that all angles of the situation are being examined. An example of geospatial intelligence preparation and exploitation is the Geospatial Intelligence Preparation of the Environment (GPE) methodology developed by the National Geospatial Intelligence Agency (NGA) (Fig. 1a). This methodology, based on the Joint Intelligence Preparation of the Battlespace (JIPB) (Fig. 1b) doctrine in use by America's military, has been modified such that it allows for the analysis and intro-



Intelligence, Geospatial, Figure 1 GPE compared with IPB [2]

duction of civilian and other non-military situations. This modified approach provides GEOINT with extreme flexibility in that the process can be used for disaster management, security planning for large events, emergency response, etc.

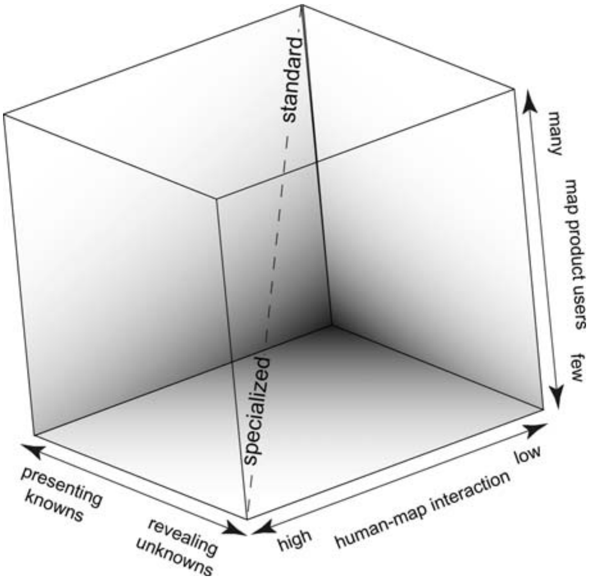
In keeping with this flexible nature, GPE is composed of four components that do not always have to occur in a strict linear fashion. The components represent key areas of the intelligence problem that provide a basic and systematic structure for evaluating any given situation and may be revisited as new information comes in. The components are:

1. **Define the Environment**: The first component of NGA's GPE methodology requires that the analyst define the environment. This first step uses basic information such as grid coordinates, vectors, latitude and longitude, altitude, natural and political boundaries, etc. to define the exact area of interest upon which the resulting GEOINT product will be built.
2. **Describe the Environment's Influence:** Once the base is established the analyst will describe the environment's influence by collecting all existing information on natural conditions, infrastructure, and culture in the area of interest. Any thing that can affect operations carried out in these regions must be considered, including: weather; transportation networks; language; cultural, political, ethnic, and religious factors and boundaries, etc.
3. **Evaluate Threats and Hazards:** In this component the analyst draws from the other INTs to evaluate threats and hazards in the area and layers this intelligence data onto the information base established in the first two components. This component includes data on the size and strength of the enemy, their tactics and doctrine, whether there is currently (or exists potential for) an insurgency in the area, any possible chemical/biological threats, etc.
4. **Develop Analytic Conclusions**: In this last component all information is integrated to develop predictive analytic conclusions. In this step the analyst attempts to determine the most likely course of action that the enemy or threat will take, and then to analyze the implications of those actions on the overall situation. During this process the analyst is constantly trying to increase his level of certainty over time by compiling and analyzing data and information until it reaches a level that allows him the confidence to raise a predictive alert or issue a report before the occurrence of an event [4].

The products created by GEOINT are wide ranging and vary in nature between what those in the field consider to be regular or standard products, and special or specialized products. Standard GEOINT products include such things as maps, charts, imagery, and digital raster and vector geospatial data. They usually support the largest population of users who do not have direct use for specialized products. Standard products are also typically of a planned and routine nature, and are designed to meet certain specifications. Specialized products are those that include the use of more advanced technologies, sensors, and often the variable of time. Specialized products at the same time allow the user to extract more detailed information in a more flexible, and sometimes more dynamic, format. Because they are often tailored to meet certain objectives, the number of users of specialized products is most often minimal.

It should be noted that not all products will fall neatly into just one of the following categories as shown in Fig. 2. Figure 2 shows a modified version of MacEachren's map use cube that helps to explain GEOINT products in general and to show the distinction between standard and specialized products.

In this conceptual space the use of a GEOINT product is shown to have three continua that form extremes in two corners but never any distinct boundaries within the space. The first continuum shows how the number of end users is typically a function of how specialized or how general the product is; with many users frequently requiring access to standard products and a smaller number of more



Intelligence, Geospatial, Figure 2 The conceptualized space of GEOINT product use (*modified from* [5])

advanced users requesting or having need for the specialized products. The second continuum defines the product in terms of the level of human to map interaction. A product that allows the user more control over changing the display properties, presentation, layers included, maps viewed, merging map displays, overlay and superposition of user data, etc. is a more specialized product than one in which all the variables are static and the user is limited in the changes they can make. The third continuum deals with the data that is being delivered to the user via the product. If the product focuses more on presenting only a few general data variables to the user with a certain message in mind (i. e. a topographic map), then the product is said to be of a more standard nature. On the other hand, products that effectively present multiple sources of information layered in such a way that the user can ask his/her own questions of the data and use it to analyze a wide array of situations are said to be more specialized. The two extremes of this conceptual space are products that provide the user with much interactivity and allow for the highest possible degree of analysis, and those that offer low amounts of interaction and simply present their data. The map product users continuum must be looked at carefully in this model. While it is true, in most cases, that more users will benefit from standard products that are simple in their presentation of data and are fairly static, a new age of dynamic products that share data rapidly across networks and allow users more and more levels of control over the presentation, display, and interaction with their data are beginning to influence a larger audience.

Key Applications

Because GEOINT allows for the layering of multiple sources of information to address any specific problem, the list of GEOINT applications is varied and extremely wide ranging. What follows are a few examples of GEOINT applications that can be further researched by visiting the NGA [6], UMC [7], and other websites listed in the recommended reading section.

Maps, Charts, Imagery

One of the most common applications of GEOINT is to build and provide paper and digital topographic, maritime, and aeronautical charts and graphics, as well as standard analyzed and unanalyzed imagery to operators across every organization.

3D and 4D Fly-Through

These 3 and 4 dimensional products are a great resource for providing commanders, planners, and other mission operators with a dynamic view of the battle space. A 3D virtual representation of the area of interest is constructed and the point of view of the observer is carried around the image along specified flight paths that are constructed to give the maximum amount of information about the layout and position of the area of interest. Other layers of information can be added to the 3D virtual world and then the whole thing can be enabled with the 4th dimension of time to recreate a given situation to enable more advanced analysis.

Data Dissemination and Fusion

With the large amount of data available, GIS provides critical infrastructure to support the collection, fusion, and rapid dissemination of geospatial and other data to a wide array of users operating around the globe.

Automated Feature Extraction

Methods are being refined to build accurate vector data of roads, building, etc. from high resolution commercial and non-commercial satellite imagery in an automated fashion. Building computer programs to do the brunt of the extraction process greatly reduces the number of manhours involved.

Automated Target Recognition

This application focuses on using specific computer algorithms developed for a variety of sensor products to automatically detect, locate, and track the change in position of targets of interest such as vehicles, missile sites, tanks, etc.

Automated Scene Description

This application uses the target detection results from the ATR process and then goes one step further by providing descriptions of the contents of any scene. An example of this would be running ATR to locate a convoy moving through the desert and then having ASD provide higher-level description that the convoy is 5 miles SW of a certain facility.

Cross Country Mobility Studies

GIS is an exceptionally capable tool for analyzing the many variables that contribute to locating suitable mobility corridors across terrain. Combining variables like slope, soil composition, wet or dry conditions, land cover density and type, etc. the analyst can determine the best paths for a variety of vehicles.

Threat Range Study

These studies can be performed in a variety of ways with the same basic premise that the analyst is delineating the space between a sensitive object and a threat to that object. If the sensitive object is a boat in harbor, then the analyst will buffer out from the boat (with distance specified by the threat capability, e. g. explosives, etc.) to define the area that must be secured to prevent harm to the vessel.

3D Urban Scene Reconstruction

Constructing virtual recreations of urban areas provides extremely valuable support to operations. 3D views provide many benefits, among which is the ability to perform viewshed analysis to locate potentially good line of sight for sniper coverage of an advancing assault, or to inversely locate an enemy sniper position. 3D urban simulations are also being used to train soldiers by providing realistic situations with multidimensional environment.

Risk, Natural Hazard, and Homeland Security Management

When large scale disasters or hazardous events occur there is initially a lot of confusion and panic that must immediately be followed by coordination and purposeful action. In much the same way that action is coordinated in battle, GEOINT can provide structure for decision making, and rapid fusion and collaboration of data that is needed to handle a large scale domestic situation.

Future Directions

One of the most current and comprehensive plans to forward GEOINT was drafted as a set of challenges to NGA by a special committee formed by the Mapping Science Committee of the National Research Council of the National Academies [3]. The challenges set forth include advancements in data mining technology; sensor fusion; spatiotemporal database management systems; visualization; data sharing with coalition forces, partners, and communities at large, etc.

Cross References

► Change Detection
► Crime Mapping and Analysis
► Data Analysis, Spatial
► Homeland Security and Spatial Data Mining
► Image Mining, Spatial
► Patterns in Spatio-temporal Data
► Temporal GIS and Applications

Recommended Reading

1. Cornell Law School Legal Information Institute: http://www.law.cornell.edu/uscode/html/uscode10/usc_sec_10_00000467----000-.html (2006)
2. NGA (National Geospatial-Intelligence Agency): Geospatial Intelligence (GEOINT) Basic Doctrine: Publication 1–0 (2006) On-line at http://www.nga.mil/NGASiteContent/StaticFiles/OCR/geo_pub1.pdf
3. National Academies of Science. Priorities for GEOINT Research at the National Geospatial-Intelligence Agency. (2006) ISBN: 0–309-65862–4
4. MacEachren, A.M., Robinson, A., Hopper, S., Gardner, S., Murray, R., Gahegan, M., Hetzler, E.: Visualizing Geospatial Information Uncertainty: What We Know and What We Need to Know. Cartogr. Geogr. Inform. Syst., **32**(3), 139–160 (2005)
5. MacEachren A.M., Taylor, D.R.F. (eds.): Visualization in Modern Cartography. Elsevier, Oxford (1994)
6. National Geospatial-Intelligence Agency (NGA) in the United States of America. http://www.nga.mil/
7. Center for Geospatial Intelligence at the University of Missouri – Columbia (UMC). http://geoint.missouri.edu/
8. Defence Intelligence Collection Group (ICG) in the United Kingdom. http://www.mod.uk/DefenceInternet/AboutDefence/WhatWeDo/SecurityandIntelligence/DIS/ICG/
9. Defense Imagery and Geospatial Organization (DIGO) in Australia. http://www.defence.gov.au/digo/
10. Gordon, M.K.: NGA's Historian Reflects on First 10 Years, Pathfinder 4(5) Sept.–Oct. National Geospatial Intelligence Agency (2004)

Interaction, Space-Time

► Temporal GIS and Applications

Interdisciplinary

► Uncertainty, Modeling with Spatial and Temporal

Interesting Pattern

► Frequent Pattern

Interestingness Measures

► Co-location Patterns, Interestingness Measures

Intergraph: Real Time Operational Geospatial Applications

PETER BATTY, IGNACIO GUERRERO
Intergraph Corporation, Huntsville, AL, USA

Synonyms

Geomedia; G/Technology; Inservice; M&S Computing; Photogrammetry; Z/I imaging; Standards, geographic data object (GDO); TerraShare; Image station; Automated vehicle location (AVL); WiMAX

Definition

For many years, Intergraph has been one of the market leading commercial companies in geospatial technologies, with a broad range of products and a presence in many vertical markets. At the time of writing, Intergraph has two divisions: Process, Power and Marine (PP&M), which focuses on Plant Design and Engineering, and related technologies; and Security, Government and Infrastructure (SG&I), which sells a range of products and services leveraging geospatial technology, using a broad definition of that term.

The GeoMedia family of products is the most widely used of Intergraph's geospatial offerings, with applications in many industries including government, transportation, military, intelligence, utilities and communications. Intergraph also offers a suite of products specifically focused on the utility and telecommunications industry: G/Technology is focused on management of complex networks, and InService is used for outage management and workforce management. Intergraph has long been a market leader in the Public Safety market, providing emergency call taking systems – which many might not think of as "GIS", but which is fundamentally a geospatial application. Its products in this area have expanded to serve the growing market for emergency response and national security. Intergraph also provides a range of products for the capture, processing and distribution of imagery and terrain data, including cameras for aerial photography, the ImageStation family of products for processing, and TerraShare for distribution.

Intergraph believes that its combination of products and experience in both traditional geospatial applications, and in the real time mission critical space of public safety and security, position the company well in the growing market for real time geospatial applications.

Historical Background

Intergraph has its origins in the 1960s US space program that led to the successful Apollo 11 moon landing in 1969. At that time Jim Meadlock worked at IBM's Federal Systems Division and was responsible for integrating the computer operations for the Saturn Launch Vehicle. In February 1969 Meadlock formed M&S Computing along with Nancy Meadlock, Terry Schansman, Keith Schonrock and Bob Thurber. Jim Taylor, who would later become Intergraph's second President and CEO, joined M&S Computing in May of 1969 as employee number 6 and "honorary" founder. In 1980 the M&S Computing name was changed to Intergraph and Intergraph became a public company in 1981 trading on the NASDAQ market under the symbol INGR. Intergraph would remain a public company until November of 2006 when it was acquired by private investors.

In the 1970s, M&S Computing engaged in contracts with NASA and US Army to convert analog missile guidance systems to digital guidance systems. As part of this activity, M&S Computing developed graphic interfaces for displaying missile trajectories. Using this experience, M&S Computing developed an interactive graphics system integrating Tektronix terminals with a digitizer tablet from Summagraphics and a Digital Equipment PDP-11 minicomputer. The first interactive graphics application was a program for NASA used for printed circuit-board design. In this application, engineers were able to manipulate onscreen graphic objects representing the circuit-board components. Based on this interactive graphics technology, in 1974 M&S computing developed a mapping system for the City of Nashville. This represented the birth of GIS at Intergraph (then M&S Computing).

During the early M&S days other seminal GIS events were taking place at Harvard University. In 1964 Howard Fisher founded the Harvard Laboratory for Computer Graphics and Spatial Analysis. Fisher received a grant from the Ford Foundations to develop SYMAP, a program for mapping using a line printer as output device. Students and researchers at the Harvard Laboratory would later on influence greatly the GIS industry. Among them was David Sinton who joined Intergraph in 1979, and Jack Dangermond, who went on to found ESRI.

In the 1980s Intergraph graphics technology evolved into "turnkey" systems integrating hardware and software. Intergraph adopted the successful Digital Equipment VAX/VMS minicomputer platform. This platform was augmented with specialized graphic processors. In this period Intergraph pioneered advances in software and hardware that brought interactive computer graphics to new levels. Intergraph introduced the Interactive Graphics Design Software (IGDS) that enabled persistent storage of graphics data in design files. In those days hard disks were not fast enough to drive display. One of Intergraph's most significant contributions was the development of a smart disk controller that was able to read graphics files (in DGN format) fast enough for interactive graphics. Intergraph systems delivered "Intergraph disks" configured at the factory for high performance. As base interactive graphics evolved so did number of CAD applications including GIS.

During the 1980s Intergraph began important developments to build data capture and cartography systems. A contract with the Defense Mapping Agency (DMA) led to the development of new generation stereo plotter; Intermap Analytic (IMA). In this timeframe Intergraph also developed a variety of specialized scanners and high resolution plotters and film writers. Notable among these was the MapSetter series.

In the late 1980s and early 1990s Intergraph shifted from mini-computer based systems to UNIX workstations. Intergraph engineering workstations were powered by advanced 32-bit RISC processors of Intergraph design. This is the time when Intergraph microprocessor intellectual property was built which later led to the patent violation lawsuit filed against Intel filed in 1997. The Intel dispute eventually led Intergraph to exit the hardware business to focus solely on software and services. The Intergraph Intel litigation culminated with a settlement which recognized the value of Intergraph intellectual property.

During the late 1980s and early 1990s Intergraph developed robust GIS systems and applications. FRAMME, targeted for the Utilities industry, introduced advanced data modeling incorporating a rules based feature model and network model. TIGRIS introduced an innovative object-oriented native topological model. MGE introduced robust CAD-based GIS focused on data capture and spatial analysis. These systems incorporated advanced technologies using proprietary data storage and proprietary object systems as other systems did at the time. The base graphic platform used for FRAMME and MGE was Microstation, a system originating in IGDS developed by Bentley Systems.

Parallel with the evolution of GIS software, Intergraph continued developing specialized mapping systems, photogrammetry in particular. In 1988 Intergraph announced the formation of Z/I Imaging, a joint venture with Carl Zeiss to build state of the art end to end photogrammetry systems. Z/I Imaging developed analog and digital mapping cameras as well as processing software. Z/I imaging complemented the photogrammetry software portfolio with TerraShare for image management. In 2002 Intergraph acquired 100% ownership of Z/I Imaging.

In the mid to late 1990s, Intergraph embarked on the development from the ground up of new GIS technology that became the base for GeoMedia and G/Technology. Several tremendously important concepts emerged at that time. First it was recognized that GIS was a database (transactional) application rather than a document (or tile) based application as it was in previous generations. Secondly, it was recognized that GIS should leverage on open IT standards rather than rely on proprietary technology. Thus GeoMedia technology is based on standard databases (such as Oracle) and standard object technology which enables enhanced flexibility and extensibility. GeoMedia pioneered data access to multiple sources with no translation. This was realized by the creation of a specification (Geographic Data Objects – GDO) which provided a public API for data access. The fundamental change was the change from a format specification to an API specification.

With the advent of the worldwide web, Intergraph leveraged the open standards based GeoMedia architecture to develop web GIS using objects in common with desktop applications.

Scientific Fundamentals

GeoMedia

GeoMedia is a broadly applicable suite of products for accessing, displaying, analyzing, and presenting spatial data. GeoMedia provides a full suite of powerful analysis tools, including attribute and spatial query, buffer zones, spatial overlays, and thematics. The GeoMedia suite of products consists of desktop products, web products and add-on applications for a variety of disciplines: terrain analysis, grid analysis, image processing, parcel management etc.

These products share a geospatial data access layer which is independent of the physical database storage. This capability, pioneered by GeoMedia, specifies a data access API or access protocol as opposed to physical storage. GeoMedia thus is able to implement "data servers" that connect a variety of physical storage sources: Oracle Locator database, ESRI shape files, MapInfo files, etc. These data sources are accessed directly in native mode with no translation. When connecting to an industry standard database like Oracle Locator, this approach results in automatic interoperability with other applications based on native Oracle Locator data storage.

Geospatial functionality is implemented as GeoMedia objects which expose a pubic API for customization and extensions. GeoMedia objects can be programmed to create extensions to functionality hosted by the GeoMedia desktop or to create web applications.

GeoMedia desktop applications offer a sophisticated event management and command system to support interactive graphic functions. A unique facility ("pipes") implements dynamic geospatial processing. Pipes are able to react in real time to database changes and re-compute analysis results. For example, if the user creates a buffer zone around a road feature and the road is edited, the buffer zone will immediately update itself. This real-time dynamic capability is complemented with a powerful expression evaluator system for attributes, geometry and style. GeoMedia web applications can be created from scratch using the public APIs or they can be generated using a publishing mechanism that derived the application look and feel from settings stored in the database.

G/Technology

There have historically been two approaches to geospatial applications in utilities and communications: one has been to apply generic geospatial products, and the other has been to develop specific geospatial products focused on the special needs of this market segment. Intergraph's G/Technology falls into the latter category, following in the tradition of other products such as IBM GFIS, which had its origins in the late seventies and was strong in the utility market in the eighties and early nineties; Smallworld, which came to prominence in the utility and communications market in the early to mid nineties, and Intergraph's own FRAMME product, which was the predecessor to G/Technology.

There are several issues specific to utilities and communication which have led to the development of products specifically focused on this market. One is the requirement to model complex networks. The traditional simple model of point, line and area features is not rich enough to effectively handle more complex network features such as switches and cabinets, which may have many connections at a single point, with complex logic to determine how items are connected. Ability to handle other types of relationships and compound features is also important for these types of systems. G/Technology has sophisticated data modeling capabilities which address all of these issues.

Utility and communications applications also present demanding challenges with regard to scalability, in terms of both database volumes and number of users. G/Technology utilizes an innovative caching technology, the Dynamic Data Cache (DDC), which enables the use of Oracle as the repository for all aspects of the data, both spatial and non-spatial, while also providing the performance benefit of a highly optimized graphical data format. Thirdly, managing many concurrent users involved in long transaction updates is important for large utilities and communications customers. G/Technology implements a highly scalable version management scheme within Oracle, which has been proven in production use with tens of thousands of versions.

TerraShare

TerraShare is an enterprise infrastructure for management of raster and elevation data, and earth imaging production. TerraShare integrates storage infrastructure with end-user production and exploitation tools, enabling individuals and organizations to address their geospatial data management, access, and distribution needs effectively. This unique solution addresses the complete geospatial raster and elevation information life cycle within the enterprise.

ImageStation and Aerial Cameras

Intergraph provides a comprehensive suite of digital photogrammetric software. The ImageStation product line includes applications for the entire production workflow. By combining Intergraph's data acquisition, data exploitation, and data distribution hardware and software systems, Intergraph provides a completely integrated, end-to-end photogrammetric workflow solution.

Intergraph's full line-up of products includes film and digital cameras, mission planning software, and flight management/sensor management systems. Intergraph's Z/I Imaging DMC (Digital Mapping Camera) is the industry's most innovative and precise turnkey digital camera system. Z/I Imaging partnered with Carl Zeiss to develop a unique lens design, minimizing distortion and maximizing resolution. The DMC supports aerial photogrammetric missions for the broadest range of mapping, geographic information systems (GIS), and remote sensing applications.

I/CAD and Security

I/CAD is Intergraph's solution for Computer Aided Dispatching, which was initially developed for the Public Safety market, to handle call taking and dispatching for police, fire and ambulance services, and which is a market leader in this space. This was one of the first applications to leverage early automated vehicle location (AVL) technologies. This platform is also used for dispatching in other markets, including automobile clubs, utilities (for both outage management and workforce management), and most recently in the growing market for security applications.

In the security market, the platform has been extended to handle integration with video cameras and other types of sensor, including access control and intrusion detection systems, and chemical, biological and radiation sensors. A key technical feature of I/CAD is its high availability capabilities, which provide redundancy in all areas of the system and have enabled many systems to run continuously for years with zero downtime.

Key Applications

Intergraph provides geospatial solutions to a wide range of industries, including commercial photogrammetry, transportation, public safety, security, local and central government, military and intelligence, utilities and communications. The following sections outline some typical applications in these areas.

National Security

Intergraph provides a range of applications to support national security, including antiterrorism and force protection initiatives. Many real-time inputs from sources such as video cameras, location tracking devices, access control and intrusion detection systems, chemical, radiation and biological sensors, and traffic sensors can be combined with other geospatial data to present a "common operating picture" to emergency responders and planners, which provides them with real time situational awareness. Applications handle "consequence management", handling complex responses plans to different situations. Intergraph solutions are employed in some of the most high profile security systems in the world.

Public Safety

Intergraph provides emergency call taking and response systems for police, fire, ambulance and other emergency response agencies. High performance and high availability are of course key aspects of such systems. Again, interoperability between multiple agencies is often an important element of emergency response systems. Intergraph also provides related applications, such as records management systems for recording and analyzing incidents, crimes and other information. Spatial analysis plays an important role in highlighting patterns in this data.

Government

Intergraph solutions are used in a wide variety of government applications, from public safety preparedness and response to land information, public works, and transportation management, spanning the complete life cycle of geospatial data – from creation to distribution. Intergraph solutions are used in managing spatial data infrastructures and creating eGovernment solutions, often leveraging OpenGIS standards which play an important role in these application areas.

Transportation

The transportation industry employs Intergraph geospatial solutions to ensure dissemination of accurate and timely information, keeping people and products moving safely and efficiently. From collecting information in the field to visualizing roadway assets and congestion on a screen, map, or via the Web, geospatial solutions are helping hundreds of state, provincial, and national governments solve their asset management and capital planning problems. Transportation operations management applications allow rail and transportation agencies better maintain accurate roadway and railway inventories to serve their constituents. These solutions can be integrated with the capabilities for incident management, command and control, and security, discussed previously, enabling personnel to see all transportation-related data in a single, common operating picture. This helps transit agencies, airports, and seaports simplify monitoring, and speed response to security-related issues.

Utilities and Communications

Managing assets, such as personnel, mobile resources, equipment, and facilities, can be one of the most time-consuming and labor-intensive aspects of any utility or communication organization. Managers require tools that streamline workflow processes to effectively meet operational requirements, budgetary constraints, and regulatory demands. Geospatial resource management applications allow organizations better manage and secure their resources – from service technicians, repair equipment, and emergency response vehicles to military field personnel and equipment – resulting in increased productivity and profitability, and improved customer satisfaction.

These infrastructure and resource management applications improve response time and improve customer service by enabling these organizations to use personnel effectively, eliminate redundant efforts, and manage assets easily and efficiently. Broad infrastructure networks of pipes, wires, cables, and fiber require management of many assets, including service centers, mobile work crews, vehicles, and facilities. By binding interrelated systems – geofacilities, outage, work, and mobile workforce management – information is instantly available across the enterprise. Through the integration of asset and maintenance data with the geospatial data of the network, these agencies and organizations can follow prioritized maintenance plans, dispatch mobile workers efficiently according to location and availability, and keep an up-to-date inventory of assets – even across expansive geographic areas.

Military and Intelligence

Geospatial intelligence exploitation and production solutions help military and intelligence professionals meet their operational goals and enable data sharing across the enterprise and around the world. Production systems for the creation and maintenance of spatial data must meet rigorous specifications, such as those defined by national intelligence agencies. Again, a trend in this area is towards combining many sources of real time data into a geospatial common operating picture.

Future Directions

Wireless Networks, Location Tracking and Sensors

With wireless network access becoming increasingly pervasive, there will be growing use of this technology in geospatial applications. There are several technologies competing to provide this service, including cell phone networks, WiFi, WiMAX and satellite-based approaches for remote areas.

Location is of significant interest for mobile applications, and location tracking devices are progressing rapidly – GPS is becoming commoditized. In addition, local tracking technologies such as RFID and ultra wideband (UWB) are gaining momentum, which open new application areas for geospatial technologies. The number of non-spatial sensors is growing phenomenally, including video cameras, traffic sensors, etc., which require a geospatial context to make sense of them.

These sources, along with others, will provide huge amounts of geospatial data, enabling the creation of many new applications which provide new technical challenges. For many applications, self-updating databases will replace the current highly manual update processes. In a few years, the technology will exist to know where almost everything is, all the time. While this technology creates enormous opportunities, it also brings challenges in handling potential privacy issues.

Real-time Geospatial Applications

All of the factors mentioned previously are contributing to considerable growth in real-time operational geospatial applications. With the recent natural disasters and terrorist incidents, security and emergency response are major focus areas for this type of application. Transportation agencies and organizations are taking great strides to protect travelers, employees and assets. Major transportation authorities are installing infrastructure protection systems that use thousands of cameras, as well as intelligent video and sensor technologies, to monitor subway and rail systems, highways, bridges and tunnels. This type of system can provide a real-time common operational picture, which provides a detailed view of what is happening in the real world.

Geospatial Technology Enters Mainstream IT

Major IT vendors such as Google, Microsoft and Oracle all now provide significant geospatial technology. Many aspects of basic geospatial functionality are becoming commoditized, and increasingly large amounts of data are becoming freely or very cheaply available. This change removes what was previously the biggest single obstacle to broader adoption of geospatial technology: the prohibitive cost of each organization capturing and maintaining its own data. The IT market is finally reaching the point where geographic information is regarded as just another data type. This trend makes it increasingly hard to look at the geospatial market as a distinct entity, since geospatial elements are being introduced into so many different applications.

This infusion of technology, talent and new ideas is excellent for the geospatial industry, but it poses some questions for traditional geospatial vendors. Those who focus on providing vertical applications to solve specific business problems are better positioned to leverage these new developments as appropriate. Vendors who focus on horizontal platform solutions face more of a direct challenge from this change in the market.

Cross References

► Photogrammetric Applications
► Smallworld Software Suite

Recommended Reading

Prendergrasr, S.: Intergraph's Jim Meadlock Reflects on 30 Years of Graphics Technology. GeoWorld (October 1999), 54–56

Dunn, J.: Intergraph at the Heights. Business Alabama **18**(8), 19–24 (2003)

Chrisman, N.: Charting the Unknown: How Computer Mapping at Harvard Became GIS. ESRI Press (2006). ISBN 1-58948-118-6

Internet GIS

DAVID MORETZ
Department of Geography, University of Minnesota, Minneapolis, MN, USA

Synonyms

Web GIS; Distributed GIS

Definition

Internet GIS can be defined as network-based geographic information services that utilize both wired and wireless Internet to access and distribute geographic information, spatial analytical tools, and GIS web services. The basic concept of a distributed GIS is a major paradigm shift from the traditional desktop GIS model, which encouraged the use and development of proprietary and all-encompassing GIS software programs and data models. With Internet GIS, the underlying design concept is radically changed to shift the focus from proprietary architectures to standards-based components with specialized functions that can interface with other components and GIService providers. This shift has altered the landscape of GIS to foster the growth of many small Internet GIS service providers and specialized services.

Historical Background

The emergence of Internet GIS is significant, as it creates the ability to more easily share and disseminate GIS data, thereby reducing the amount of redundant data collection and creation. The ability to access GIServices over the Internet also expands the reach of GIS to organizations and governments that previously did not have the capacity, funds, and/or skillsets to implement full GIS capabilities previously. Finally, with the development of Internet GIS, applications are being developed that target less sophisticated GIS users and function to broaden the awareness of and provide practical applications that have mass appeal and provide useful benefits to everyday human activities. Examples of some of these applications include Google Earth, MapQuest, and OnStar.

Underlying the concept of Internet GIS is the issue of data publishing and distribution. Internet GIS implies the distributed nature of data and therefore allows data providers to maintain their data directly at the source, ensuring the timely updates and version controls that are necessary to maintain data integrity and accuracy. At the same time, Internet GIS enables service providers to remotely access the source data warehouses directly over the Internet without having to download and continually update large datasets. In order to support the evolving development and proliferation of Internet GIS, the fundamental framework of geographic information systems has to be altered to accommodate this shift. Included in the re-architecting of GIS is the issue of the supporting technology infrastructure, which includes network and software configurations to enable a Client/Server architecture. Other issues to be considered are the structure and nature of data and the necessity for data standards to enable integrated GISystems. And finally, with the introduction of Internet GIS, the landscape and functionality of the basic GIS applications change. Internet GIS introduces the concept of Web Mapping and distributed GIServices that enable specialization and integration with GISystems from around the world.

Scientific Fundamentals

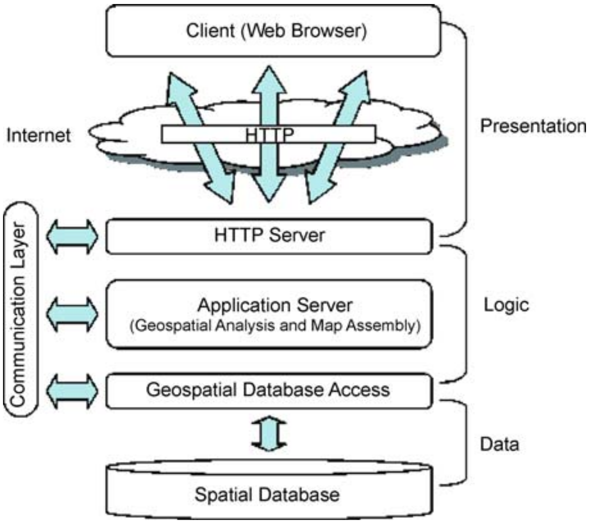
As the name implies, Internet GIS relies on the wired and wireless Internet for users to communicate or interact with spatial data storage and processing facilities. This contrasts with the concept of desktop GIS where the user interacts with spatial data and processing facilities via a GIS application and data stored locally or accessed over a LAN (Local Area Network).

This difference has significant implications in terms of how data is stored and processed, because the nature of a distributed environment is inherently more complex. Given the fact that spatial data is significantly larger than non-spatial data, the complexities of Internet GIS are magnified. Examples of some of the unique aspects of spatial data include the shear size of the data sets and the complex indexing and filtering of data needed to enable efficient transfer of data over a network. Spatial data sets often are measured in terms of Terrabytes instead of Megabytes, and relationships between the data are based on spatial components such as orientation to and distance from. The following sections describe some of the unique attributes of the Internet GIS Architecture.

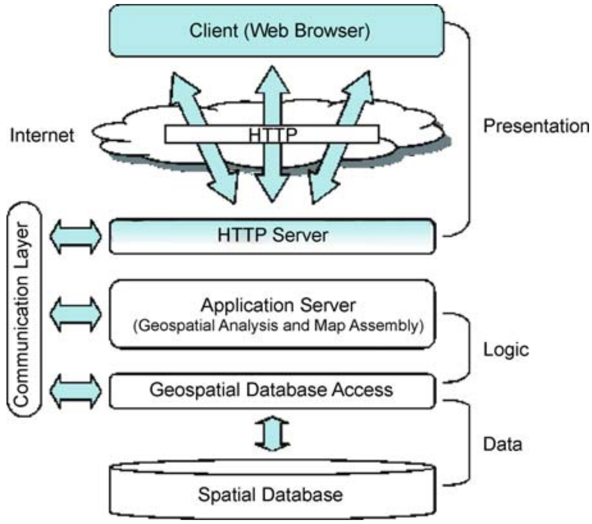
Internet GIS Architecture

Most GIS applications utilize the client/server model. The basic concept behind this computing model is that one element makes a request (the client) and another element fulfills that request (the server). An analogy can be made to a restaurant where a customer requests an entrée from the menu and the waiter returns with the desired entrée. The typical client/server application consists of three primary elements: presentation, logic, and data. The presentation represents the user interface, the logic represents the processing, and the data refers to the database or database management system. In the restaurant analogy, the presentation would be the menu, the logic would be the process of the waiter placing the correct order with the chef, and the data would be the actual entrée.

When this model is applied to Internet GIS, the simplistic view is of a three-tiered system in which the web browser represents the client, the web server represents the logic, and the database or database management system represents the data. For example, a person would use their web browser to enter the URL that contains the spatial data (or more practically, a map) that is desired. The web



Internet GIS, Figure 1 Basic presentation of Internet GIS architecture and breakdown of the three-tiered structure



Internet GIS, Figure 2 Internet GIS architecture showing the partition for distributed presentation

browser would then connect to the specified web server via the Internet to ask the server for that particular data. The web server would then transfer that request to the database management system. The database management system would then return the requested data to the web server. The web server would in turn reformat the data into a format that the web browser could understand, and then send the data back to the web browser in the form of HTML.

Conceptually, Internet GIS architectures typically consist of multiple layers. While the three-tiered structure tends to provide a high-level model for understanding the basic components of Internet GIS, the actual systems that implement the logic and data tiers of the architecture tend to be more complex.

Diagram 1 is a basic model for the architecture logic behind an Internet GIS.

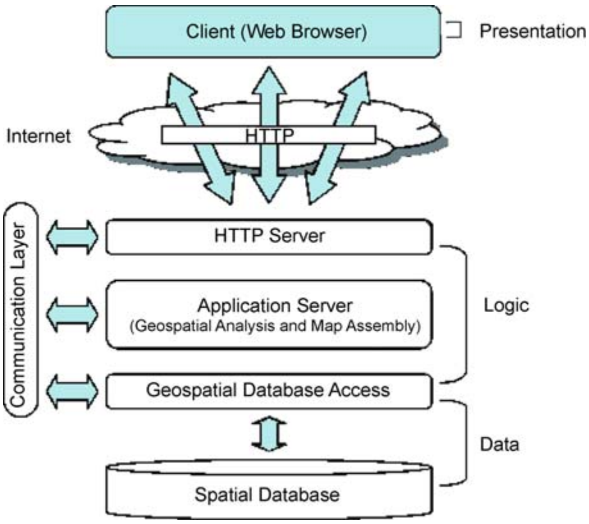
The capability and flexibility of Internet GIS becomes interesting when the components that make up the architecture are considered as individual components that can physically reside on any computer in the world. Internet GIS systems can access data warehouses directly without having to download and store separate datasets. Internet GIS systems can also integrate with other Internet GIS systems at the Logic tier to utilize distributed GIServices. Through APIs, it is possible to access geospatial analysis tools and processes that reside within other systems. These capabilities open up a whole host of possibilities for broad distribution and access to geospatial data and analysis.

While the three-tiered application elements – presentation, logic, data – represent clean partition points, complexities and greater functionality and flexibility can be gained by moving the partition points to the boundaries between the elements. Any combination of these partition points can and usually are implemented in Internet GIS depending on the desired application and infrastructure. The following is a description of the various partition points and their implications:

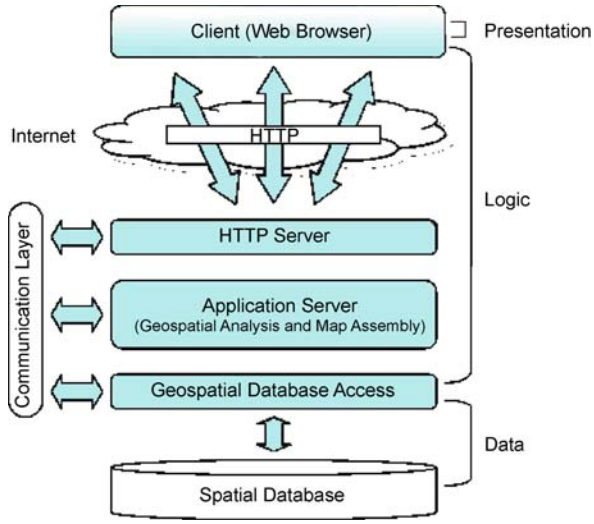
Distributed presentation *[partition point on presentation element]:* A very thin client configuration in which the data, logic, and even part of the client resides in the server environment. The client is responsible for presenting only a portion of the interface or a mirror of the server environment. A simple Web browser that has no plugins or applets and only renders HTML is another example of a distributed presentation.

Remote presentation *[partition point at boundary between presentation and logic]:* The entire presentation functionality is on the client side, while the logic and data are on the server side. An example is a server-based Internet GIS that uses CGI to process user requests on the server. The results are then presented to the web client.

Distributed function *[partition point on logic element]:* The distributed function partition splits the logic element between the client and the server and puts the presentation on the client machine. It allows the client to execute some functions while it sends other more complex functions to the server for processing. Frequently, Internet GIS that use Java applets or ActiveX controls fit into this category. Examples of the basic functions performed on the client machine include query, zoom and pan, while functions such as address matching and image analysis are performed on the server.



Internet GIS, Figure 3 Internet GIS architecture showing the partition for remote presentation



Internet GIS, Figure 4 Internet GIS architecture showing the partition for the distributed function

Remote data access *[partition point at boundary between logic and data]:* The remote data access partition puts presentation and application logic on a client that retrieves data from a remote database. This architecture is called a thick client, meaning the client is responsible for all logic operations. An example is an Internet GIS that uses SQL APIs to make calls directly to a relational database.

Distributed database *[partition point on data element]:* The distributed database partition splits the data management function between the client and one or more servers, while allocating the logic and presentation elements to the client.

Distributed GIS Standards

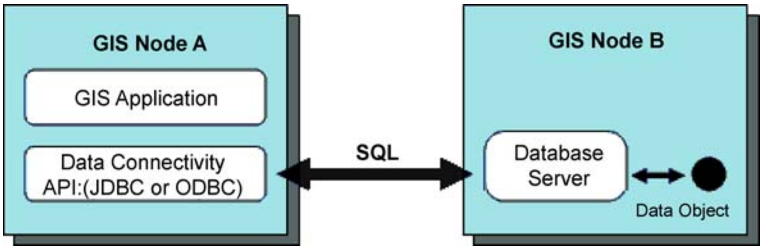
While Internet GIS creates the possibility of fully integrating geospatial data and geoprocessing resources into mainstream computing, the challenge resides in developing a widespread infrastructure of interoperable geoprocessing software and geodata products. To address this challenge, the Open GIS Consortium was founded by members of both industry and academia in 1994 to develop a framework for software developers to create software that enables users to access and process geographic data from a variety of sources across a generic computing interface within an open information technology foundation.

The framework created by the Open GIS Consortium is called the OpenGIS specification. The OpenGIS specification includes an abstract specification and a series of implementation specifications for various DCPs such as CORBA, OLE/COM, SQL, and Java. Developers use OpenGIS conformant interfaces to build distributed GIServices, which include middleware, componentware, and applications.

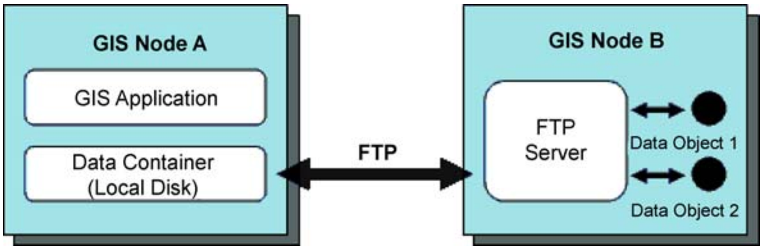
The OpenGIS specification is based on three conceptual models: the Open Geodata Model (OGM), OpenGIS services, and the Information Communities Model. The OGM specifies a common data model that uses object-based and conventional programming methods to digitally represent the earth and earth phenomena mathematically and conceptually. OpenGIS services defines the set of services needed to access and process the geodata defined in the OGM and to provide the capabilities to access, manage, manipulate, represent, and share the geodata with the GIS community. The Information Communities Model employs the OGM and OpenGIS services in a scheme to automatically translate data between different geographic feature lexicons, facilitating the collaborations among different GIS research domains and applications.

In order to successfully organize, maintain, and transfer data between organizations and systems, a geospatial metadata standard is essential. In the development of a distributed-network environment, the use of metadata plays a vital role for the interoperability of heterogeneous systems and data models. The conceptual model for geospatial metadata includes information regarding the description, history, and findings of the data. The description focuses on generic features of the data; the history refers to the derivation, update, and processing chronology of the datasets; and the findings consist of aspects such as precision, consistency and accuracy.

While there are several variations of metadata standards, the two most commonly used standards are the FGDC and ISO standards. Both standards require hundreds of fields to be filled out completely and focus on components such as identification, data, quality, spatial data organization, spa-



Internet GIS, Figure 5 Data connection representation for Internet GIS with remote data access



Internet GIS, Figure 6 Data connection representation for Internet GIS with a distributed database

tial reference, entity and attribute, and distributed information.

Geography Markup Language

To enable geodata interchange and interoperability over the Internet, a standard data structure is needed. Geography Markup Language (GML) has been designed to define this structure. GML is a markup language based on the XML standard to construct structured spatial and nonspatial information to enable data sharing and interchange over the Web. It offers a standard way to encode spatial features, feature properties, feature geometries, and the location of the feature geometries based on a standard data model.

GML supports interoperability among different data models by providing standard schemata and metadata to describe the geospatial data. When GML-coded geospatial data are transported, all the markup elements that describe the spatial and nonspatial features (geometry, spatial reference, etc.) are transported to the recipient. Based on the standard markup elements, the recipient knows exactly what each data component means and how to process that data so that nothing gets lost or distorted. The downside of GML, however, is the fact that the detailed description of GML feature elements cause the GML coded data to be very large. The large size of GML is currently a problem for transporting it over the Internet, but compression methods are being studied to try to reduce this current constraint.

Key Applications

Internet GIS has a wide range of applications and uses. Examples of these applications can be arranged into general groups such as data sharing and dissemination, data processing, and location-based services, and Intelligent Transportation Systems. The following is a list of some of these applications and a brief description regarding their use.

Data Sharing and Dissemination

Internet GIS is the ideal mechanism for data sharing and exchange over the Internet. Not only can simple raw data be distributed through FTP, but it can also be searched for and used as if it were local. For example, if a person has a need to obtain information about the Connecticut River, it is possible to access a GIS clearinghouse or data portal to find and download the necessary data. Because of the emerging data standards, this data is in a format that is usable for all users despite their chosen software application.

Online Data Processing

Traditionally, data analysis was only available to GIS professionals who had access to expensive and complicated GISystems. With the advent of the Internet and therefore Internet GIS, many data processing procedures are now available over the World Wide Web. Popular websites such as Mapquest and Yahoo! Maps in particular allow users to perform basic spatial network analysis such as shortest path queries (general driving directions from point A to point B) and buffers (all of the Chinese restaurants within 10 miles of point A). Also, commercial software such as Google Earth provides users with easy access to spatial data from all over the world.

Location-Based Services

Location-Based Services refer to a system that provides real-time data about a location and its surrounding areas. This technology is dependent on Internet GIS and mobile devices that allow users to access and interact with data at precise locations. Examples of some location-based services are access to real-time traffic information and driving directions from where the user is to locations of nearby businesses and attractions.

Intelligent Transportation Systems

Intelligent Transportation Systems link Internet GIS and real-time traffic information to provide and disseminate real-time travel information to end users. Several local governments have implemented ITS to enable users to plan travel itineraries based on real-time traffic conditions. An example of such a system is the one King County, Washington developed to improve access and usability of its public transportation system. This system provides predictive capabilities for users to find out when the next bus is coming, as well route planning functionality to determine the best way to reach a desired location.

Future Directions

The adoption, migration, and implementation of Internet GIS is filled with exciting possibilities. The hope is that increased accessibility to spatial data and analytical tools will enable scientists to build more realistic models to solve research problems and focus on the domain of the problems instead of the mechanisms of system implementation. The goal of Internet GIS is to encourage and enable the use of geographic information to make more informed decisions. While there has been much progress and research conducted so far, there is much more to do. Coordination and collaboration are needed to more fully develop a data infrastructure that enables easy searches for data and assurances of data quality and accuracy. More fully developed data standards are also so that the concept of seamless data integration can be realized. Finally, more sophisticated compression methods, indexing strategies and processing techniques are needed to successfully deal with the massive amounts of data and the taxing storage and transmission issues that result. The potential of Internet GIS is great and so is the hope that it can improve the quality of everyday human life.

Cross References

► Internet-Based Spatial Information Retrieval
► University of Minnesota (UMN) Map Server

Recommended Reading

Peng, Z.R.: Internet GIS: distributed geographic information services for the Internet and wireless networks. Peng, Z.R., Tsou, M.H. (eds) 1st edn., vol. 36. John Wiley & Sons, Hoboken, NJ, USA (2003)

Shekhar, S., et al.: WMS and GML based Interoperable Web Mapping System. GIS 01. Atlanta, Georgia (2001)

Shekhar, S., and Chawla, S.: Spatial Databases – A Tour. Horton, M. (ed). Pearson Education, Inc., Upper Saddle River, NJ, USA (2003)

Sonebraker, M., Rowe, L.: The Design of PostGRES, vol. 1. Association of Computing Machinery, Berkeley, CA, USA (1986)

Internet Mapping Service

► Web Feature Service (WFS) and Web Map Service (WMS)

Internet-Based Spatial Information Retrieval

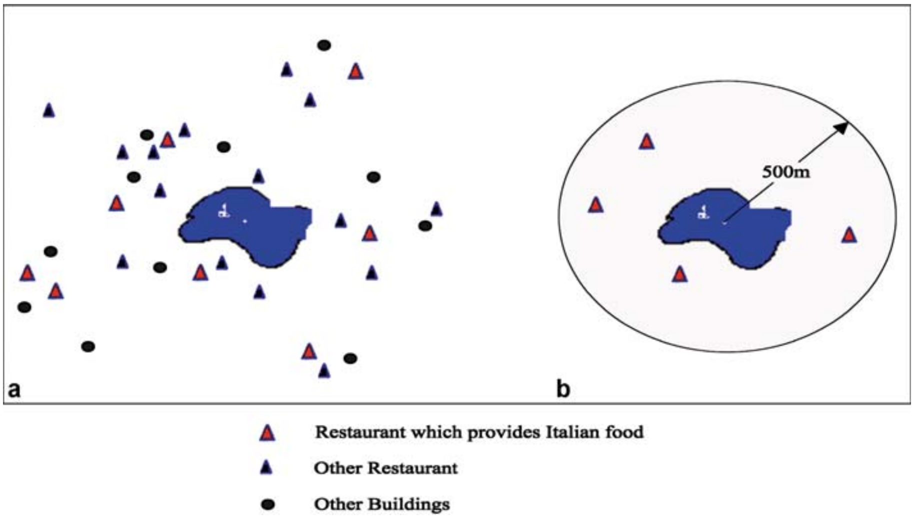
WENWEN LI[1,2], PHIL YANG[1], BIN ZHOU[1,3]
[1] Joint Center for Intelligent Spatial Computing, and Earth System and GeoInformation Sciences, College of Science, George Mason University, Fairfax, VA, USA
[2] Institute of Remote Sensing Applications, Chinese Academy of Sciences, Beijing, China
[3] Department of Electronic Engineering, Tsinghua University, Beijing, China

Synonyms

Information retrieval; Geographical information retrieval; Keyword Search; Text Search

Definition

Spatial information retrieval (SIR) refers to providing access to georeferenced sources by indexing, searching, retrieving, and browsing [1]. SIR is an interdisciplinary topic involving geospatial information science, data mining, computer networks, cognition, and cartography. An example of SIR query could be "Where are the Italian restaurants within 500 meters of a specific park?" Figure 1a illustrates the spatial distribution around the park: the shaded area represents the park; triangles represent restaurants; the red triangles represent those providing Italian food. To answer the question, spatial entities, such as restaurants and the park, are abstracted to a point described in [x, y] coordinates. To execute the query, both an information retrieval algorithm and spatial relationship analysis are needed: (1) locate the area of interest where a circle has



Internet-Based Spatial Information Retrieval, Figure 1
A query example of spatial information retrieval

a radius of 500 m (Fig. 1b), (2) filter out restaurants from other objects (represented by the point in Fig. 1), (3) search the restaurant index to identify the keyword "restaurant types" with "Italian food", (4) send a list of restaurants, ranked by the distance to the park, back to users.

Spatial Data Sources

Spatial retrieval procedures may be categorized by the types of spatial data sources, such as a digital library and the world wide web.

SIR in Digital Library In general, a digital library stores a large volume of categorized information, for example, water resources below ground in a valley for flood analysis and monitoring. Researchers [2] proposed an automatic geospatial information processing system to provide fast access to a digital library. In such a system, Geographic place names (terms) and their attributes are extracted and identified based on a thesaurus and semantic information containing spatial relationships, such as "adjacent to a lake", "south of the river". Geographical coordinates are retrieved and probabilistic weights are assigned to the place names based on their occurrence in the thesaurus. Therefore, each term can be denoted as a three-dimensional (3D) object, dimensions $[x, y]$ represent geographic coverage, and dimension Z represents weight of the term. Finally, all the terms extracted are denoted in 3D space to form a "skyline", where weights are summed when two terms have overlaps in geographical coverage. The geographic area where the peak of the "skyline" located will be indexed. Then, by applying the algorithm to all texts stored in the digital library, the entire index can be established to assist fast access.

SIR in the World Wide Web The context is complex in internet-based SIR because the world wide web (WWW) contains a huge amount of information. For example, as estimated by Danny Sullivan in 2005, Google indexed more than 9 billion documents in its crawler, where the web documents collection is built and maintained through crawling. Meanwhile, performance stood out as an issue for spatial indexing in such a large collection. To support spatial indexing in a large amount of documents, researchers [3] proposed a spatiotextual indexing algorithm with the help of geographical ontology. Each document containing place names is associated with one or more "footprints" (using coordinates to present a place) derived from ontology entries [3]. Then a three-step algorithm is applied: (1) all the documents are divided into several cells (S_i, $i = 1, 2 \ldots m$), based on their spatial distribution marked in the footprint; (2) the document sets (D_j, $j = 1, 2 \ldots n$) indexed by key words are intersected with each space cell (S_i) to form the spatiotextual index (S_i, D_j), because an index with this structure can be exploited by first searching for a textual term; then (3) the associated spatial index of documents is used to filter out those meeting the spatial constraints [3] so that the ambiguity of terms can be eliminated and query accuracy and performance is improved.

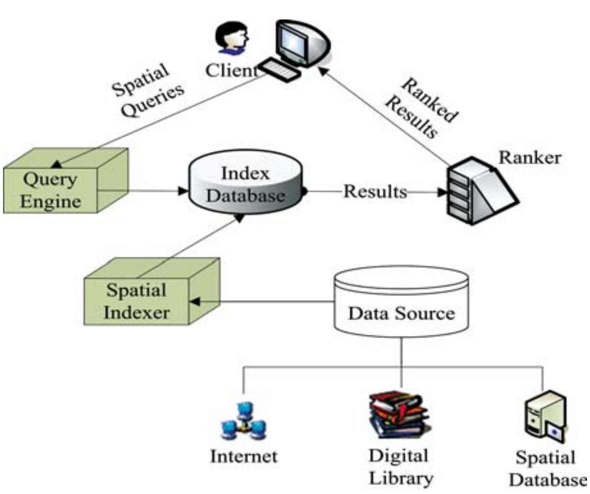
Historical Background

The history of SIR can be traced back to the year 850, when the first print book was created in China that changed the traditional mechanism of information storage. The second leap was in 1946, when the first electronic computer transformed data into a digital version. In 1950, the term "information retrieval" was first used by Vannevar Bush, and became popular [4]. In 1996, Ray R. Larson coined "spatial information retrieval" in the context of a digital library service [1]. Many SIR models have been designed and applied to retrieve geospatial information [5,6].

Scientific Fundamentals

In general, an SIR system deals with spatial queries such as "What is here?" asking for place names, geographical features about a location, or "Where is it?" resulting in a reference in a map [7]. Five types of spatial queries are summarized [1] as: *point-in-polygon query*, which is a relatively precise query asking about the geographic information of a point (denoted by [x, y]) within a area (denoted by the polygon); *region query*, asking for any geographical element that is contained in, adjacent to or overlaps the region defined; *distance and buffer zone query* (Fig. 1) refers to finding spatial objects that are within a certain distance of an area; *path query*, which requires the presence of a network structure to do shortest-path or shortest-time planning; and *multimedia query*, which combines both georeference processing and nongeoreference processing, such as pattern recognition, in executing a query.

A general SIR model for answering previous spatial queries includes the data source, a spatial indexer, a query engine, and a ranker (Fig. 2). Spatial information is obtained from geospatial data. The data are in different formats, such as textual, numerical, graphical, and multimedia. A spatial indexer provides spatial indexing by extracting geographic locations in a text or mapping data to a term based on a certain geospatial ontology [8,9]. A query engine handles user requests [10]. To provide a better quality of service, a ranker is normally used to sort the results based on match level.



Internet-Based Spatial Information Retrieval, Figure 2 A general model for spatial information retrieval

Geospatial information sources for spatial retrieval are generally available from a digital library or the WWW, where large amounts of data are stored in a variety of formats, including basic text documents, airborne and satellite images, maps from specific geographic locations, and other forms. Therefore, it's of great importance to extract and index the spatial element from data sources to improve query efficiency [11,12]. Meanwhile, the dynamic, incoherent nature of the WWW makes it more difficult for a SIR system to gather information or make spatial indexing structures scalable and efficiently updatable. The solutions to these problems relying on probability theory and spatial reasoning as discussed in "Key Applications".

Key Applications

SIR is widely used in applications ranging from scientific research and government planning to daily life.

Earth and Planetary Research

Terabytes of imagery data are collected through satellite and airborne remote sensors every day [13].While providing valuable resources to earth system research, the imagery also complicates the management of the data. SIR could help to get the data with the appropriate spatial characteristics, time span, and format from a vast number of datasets. Project Sequoia [6] gives a successful example of earth scientists being able to retrieve information from tens of terabytes of spatial and geographical data.

Disaster and Disease Analysis

SIR can assist researchers and policy makers to extract emergency information, for example, the asset losses of cities during a river flooding [14,15].

Urban Transportation

SIR can be applied to urban transportation management by indexing and analyzing the transportation datasets.

Environment Protection

The US Environment Protection Agency (EPA) has its Aerometric Information Retrieval System and national Pesticide Information Retrieval System for viewing and researching the spatial distribution and trends for pollution or other environment destructions [16].

Traveling

SIR can be integrated in mobile devices to help travelers in guidance and route planning [1,17].

Future Directions

Any query related to location needs the support of SIR systems. Future SIR systems will be more intelligent and be able to answer questions given in natural language. Meanwhile, it will play a more important role in contemporary geospatial processing, especially in the context of the WWW.

Cross References

- ► Geospatial Semantic Integration
- ► Geospatial Semantic Web
- ► Geospatial Semantic Web: Applications
- ► Geospatial Semantic Web, Interoperability
- ► Geospatial Semantic Web: Personalisation
- ► Indexing, Hilbert R-tree, Spatial Indexing, Multimedia Indexing

Recommended Reading

1. Larson, R.R.: Geographical information retrieval and spatial browsing. In: Smith, L.C., Gluck, M. (eds.) Geographical Information Systems and Libraries: Patrons, Maps, and Spatial Information, pp. 81–124. Publications Office, Graduate School of Library & Information Science, University of Illinois at Urbana-Champaign, Champaign, IL, USA (1996)
2. Woodruff, A.G., Plaunt, C.: GIPSY: geo-referenced information processing system. J. Am. Soc. Inf. Sci. **45**, 645–655 (1994)
3. Jones, C.B., Abdelmoty, A.I., Finch, D., Fu, G., Vaid, S.: The SPIRIT spatial search engine: architecture, ontologies and spatial indexing. In: Freksa, C., Egenhofer, M.J., Miller, H.J. (eds.) Proceedings of the Third International Conference on Geographic Information Science, Adelphi, MD, USA, 2004. Lecture Notes in Computer Science vol. 3234, pp. 125–139. Springer, USA (2004)
4. Fielden, N.: History of Information Retrieval Systems & Increase of Information Over Time. In: Biennial California Academic & Research Librarians (CARL) Conference, Monterey, CA, US 10 May 2002
5. Jones, C.B.: Spatial information retrieval and geographical ontologies: an overview of the spirit project. In: Proceedings of ACM SIGIR conference on Reasearch and developement in information retrieval, Tampere, Finnland, 11 Aug 2002
6. Chen, J., Larson, R.R., Stonebraker, M.: Sequoia 2000 object browser. In: Digest of Papers, COMPCON Spring 1992. The 37th IEEE Computer Society International Conference, Los Alamitos, US (1992)
7. Oyvind, V.: Geographic Information Retrieval: An Overview. Internal Doctoral Conference, IDI, NTNU, Norway (2004)
8. McCurley, K.S.: Geospatial Mapping and Navigation of the Web. In: Proceedings of the 10th International Conference on the World Wide Web, Hong Kong, 1–5 May 2001
9. Abdelmoty, A.I., Smart, P.D., Jones, C.B., Fu, G., Finch, D.: A critical evaluation of ontology languages for geographical information retrieval. J. Visual Languages Comput. **16**, 331–358 (2005)
10. Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A., Raghavan, S.: Searching the web. ACM Trans. Internet Technol. **1**, 2–43 (2001)
11. Buyukokkten, O., Cho, J., Garcia-Molina, H., Gravano, L., Shivakumar, N.: Exploiting geographical location information of web pages. In: ACM SIGMOD Workshop on The Web and Databases, Philadelphia, Pennsylvania, USA (1999)
12. Mountain, D., MacFarlane, A.: Geographic information retrieval in a mobile environment: evaluating the needs of mobile individuals. J. Inf. Sci. (2007, in press)
13. Emerson, C.W., Quattrochi, D.A., Lam, N.S.: Spatial metadata for remote sensing imagery. In: Proceedings of the 4th Annual Earth Science Technology Conference (ESTC), Palo Alto, CA, USA (2004)
14. EM-DAT Emergency Disasters Database. http://www.em-dat.net/links/disasterdbs.html. Accesed 20 Aug 2007
15. Lu, W., Mannen, S., Sakamoto, M., Uchida, O., Doihara, T.: Integration of imageries in GIS for disaster prevention support system. In: Proceedings of the XXth ISPRS Congress, Istanbul, Turkey 2004
16. Air Release (AIRS/AFS). http://www.epa.gov/enviro/html/airs/index.html. Accesed 20 Aug 2007
17. Clough, P.: Extracting metadata for spatially-aware information retrieval on the internet. In: Proceedings of the 2005 Workshop on Geographic Information Retrieval, Bremen, Germany 2005
18. Beaujardiere, J.: Web Mapping Service, OGC Implémentation Spécification http://www.portal.opengoespatial.org/files/?artifact_id=14416(2004). Accesed 9 Aug 2007

Interoperability

- ► Geography Markup Language (GML)
- ► Geospatial Semantic Integration
- ► Metadata and Interoperability, Geospatial
- ► OGC's Open Standards for Geospatial Interoperability

Interoperability, Technical

- ► Metadata and Interoperability, Geospatial

Interoperability, XML Schema

- ► Metadata and Interoperability, Geospatial

Interpolation

- ► CrimeStat: A Spatial Statistical Program for the Analysis of Crime Incidents

I

Interpolation of Continous Geofields

► Geosensor Networks, Estimating Continuous Phenonena

Intertial Motion Unit (IMU)

► Evolution of Earth Observation

Inverse Distance Weighting

► Constraint Databases and Data Interpolation

ISO

► Geography Markup Language (GML)
► Spatio-temporal Query Languages

ISO 19115

► Metadata and Interoperability, Geospatial

ISO/IEC

► Oracle Spatial, Geometries

ISO/TC 211

► Application Schema
► deegree Free Software

Isometric Color Bands Displays

► Visualizing Constraint Data

Java

► deegree Free Software

Java Conflation Suite (JCS)

► Open-Source GIS Libraries

Java Topology Suite (JTS)

► Open-Source GIS Libraries

Journey to Crime Analysis

► CrimeStat: A Spatial Statistical Program for the Analysis of Crime Incidents

K-Anonymity

► Privacy Threats in Location-Based Services

Keyword Search

► Internet-Based Spatial Information Retrieval

K-Nearest Neighbor Query

► Nearest Neighbor Query

Knowledge Based Systems

► Decision Support Systems

Knowledge Representation, Spatial

ROBERT RASKIN
Jet Propulsion Laboraty,
California Institute of Technology, Pasadena, CA, USA

Synonyms

GeoOntologies; XML triple

Definition

The formal definition of *knowledge* can be elusive. We tend to recognize knowledge when we encounter it, but have greater difficulties in defining it.

Knowledge can be distinguished from *data* and *information* [1]. *Data* are raw observations made through sensory or instrumental measurements. Spatial data typically consists of records of where something is located in space. *Information* is a summarized description of the data. In this transformation, the volume of the data decreases, but its density (value per unit) increases. Spatial information might be a mathematical representation of how an attribute varies over geographic space.

Knowledge is the next step in this chain – it includes a *context* for the ingested information. Knowledge implies that some background information is present, along with some means to build upon that information. Spatial knowledge is the understanding of where and why people or objects are located relative to one another. A GIS provides knowledge of the spatial domain through agreed upon representations of spatial coordinate systems, administrative boundaries, place names, etc. Community GIS data model extensions such as ArcHydro provide further enhancements to knowledge using hydrological concepts provided by that community.

Formal *representation* of spatial knowledge differs from the corresponding representation of data, as it must capture both facts and context. The context corresponds roughly to the common sense learned knowledge that the brain develops over time. ontologies have emerged as an effective machine-readable language for capturing knowledge. An ontology is a description of concepts and how they relate to one another, typically expressed in an XML format.

Main Text

Geographers have long studied cognitive concepts of space – how people perceive space and how these perceptions relate to actual physical space. The resulting approximations of spatial relationships capture the process of summarization and contextualization of our knowledge capture processes.

Spatial knowledge capture was integral to the early days of artificial intelligence in robotics applications. The key to making a robot functional and useful is to have it understand the nature of objects and obstacles it encounters in its path. More recently, geographers have encoded spatial cognitive knowledge directly into ontological frameworks [2,3].

An ontology is useful as a formal representation of knowledge that both humans and computers can process and append to. In the most common convention, the ontology consists of an exhaustive collection of “triple” assertions (or facts) of a subject-verb-object form such as:

Lake contains Water
North isPerpendicularTo East
Polygon consistsOf Points
AtlanticOcean isAnInstanceOf Ocean
Building isA ManMadeObject

Some of the verb forms are predefined in the language to support common understanding, while others may be user-created. The richness of the language determines its expressiveness; OWL is the currently accepted standard ontology language [4] as specified by the W3C standards body.

Spatial knowledge representation can be used to support automated processing services such as the *geospatial semantic web*. In a semantic web, concepts on web pages are understood by common search tools, such as Google. These tools enable free-text searches such as “gymnasiums in Seattle” to be translatable into maps with place-based services without further human intervention.

Spatial knowledge provides the potential for spatial reasoning. Computer chess programs use context (the state of the board) and information (the mobility rules of chess pieces) to reason about the optimal next move. It is hoped that personal agents can play an analogous role to support personal daily travel decisions.

Cross References

► Geospatial Semantic Web

Recommended Reading

1. Ackoff, R.L.: From Data to Wisdom, J. Appl. Syst. Anal. **16**, 3–9 (1989)
2. Casati, R., Smith, B., Varzi, A.C.: Ontological Tools for Geographic Representation. In: Guarino, N. (ed.) Formal Ontology in Information Systems, pp. 77–85. IOS Press, Amsterdam (1998)
3. Smith, B.: Formal ontology, common sense and cognitive science. Int. J. Hum.-Comput. Stud. **43**, 641–667 (1995)
4. W3C OWL Ontology Language Reference. http://www.w3.org/TR/owl-ref
5. Raskin, R.G., Pan, M.J.: Knowledge representation in the Semantic Web for Earth and Environmental Terminology (SWEET). Comput. Geosci. **31**, 1119–1125 (2005)

Knowledge Synthesis

► Uncertainty, Modeling with Spatial and Temporal

Knox Test

► CrimeStat: A Spatial Statistical Program for the Analysis of Crime Incidents

Kohonen Map

► Self Organizing Map (SOM) Usage in LULC Classification

Kriging

ROBIN DUBIN
Department of Economics, Weatherhead School of Management, Case Western Reserve University, Cleveland, OH, USA

Synonyms

Optimum interpolation; Optimal prediction; Variogram; Spatial dependence

Definition and Historical Background

Kriging is a spatial interpolation technique which has applications in spatial prediction and automatic contouring. It was developed by Georges Matheron, who named the technique after D.J. Krige, a South African mining engineer who did some of the early work on the topic.

Scientific Fundamentals

In a standard statistical analysis, all of the data are assumed to come from the same distribution, and can thus be used to estimate the parameters of this distribution. The idea of repeat sampling also plays an important role. For example, the reason that sample estimates are random variables is that different samples will produce different estimates of the same parameter. In spatial statistics, a somewhat different paradigm is required. In a spatial setting, often the very act of collecting the sample destroys it, so that repeat sampling is not an option. For example, if a soil sample is collected from a particular time and location, it is gone and cannot be sampled again. Spatial statistics are based on the idea of regionalized variables. The idea is that a regionalized variable has a probability distribution at each site. Thus, the realization at each sampled location is a random variable, however, the draw is from the distribution at that site and not from a global distribution for the study area. In order to make statistical inference possible, it must be assumed that the distributions share some common properties. Denote the realization of a regionalized variable at site s as $Y(s)$.

The assumption that links the distributions of the regionalized variables together is stationarity, particularly second order stationarity. If the data are second order stationary, then the local distributions share a common constant, mean and variance. The covariance is assumed to depend on the separation distance between the observations. Second order stationarity implies that the variance is finite, which may be a strong assumption in some applications. In these cases, the intrinsic hypothesis can be assumed to hold instead. The intrinsic hypothesis states that the first and second moments of the *difference* between regionalized variables, i. e., $Y(s+h)-Y(s)$, depend only on separation distance (h) and not on location (s)[1]. If the variance is finite, there is no difference between assuming second order stationarity and the intrinsic hypothesis. However, the intrinsic hypothesis is more general since it can be applied to distributions in which the variance does not exist.

If second order stationarity can be assumed, then the correlations can be represented with a correlogram which shows the correlations as a function of separation distance. When $h=0$, the correlation is one, because this is the correlation of the observation with itself. However, it is possible that the correlogram is discontinuous at the origin because, at a very small separation distance, the correlation may be much smaller than one. Matheron called this discontinuity the "nugget" effect. Figure 1 illustrates three commonly used correlogram functions (Negative Exponential, Gaussian and Spherical). Note that a nugget effect is present since the intercept is shown at 0.7. The correlograms show the "range" of the spatial relationships. This is the distance at which the correlations drop to zero (or some small number) and would be about 12 distance units for the Gaussian correlogram. This means that observations which are separated by a distance of more than 12 units will be independent.

If the intrinsic hypothesis is used, then the spatial relationships are represented by means of the variogram. The variogram is given by

$$2\gamma(h) = \mathrm{var}(Y(s+h) - Y(s))\,. \tag{1}$$

Figure 2 shows the semivariograms for the same cases as in Fig. 1 (the semivariogram is the variogram divided by two). Since the variogram is expressed as the variance of the difference between two regionalized variables, when these distributions are independent, the semivariogram will return the common variance (recall that $\mathrm{var}(x-y)=\mathrm{var}(x)+\mathrm{var}(y)-\mathrm{cov}(xy)$, in which x and y are random variables). Thus, the variogram starts at 0 (unless a nugget effect is present) and reaches its maximum at the variance (often referred to as the sill). Figure 2 shows a nugget effect since the intercept occurs at 0.3. If the variance is finite, the correlogram and the variogram are related in the following manner: $\gamma(h)=\sigma^2(1-k(h))$, where σ^2 represents the variance and $k(h)$ is the correlation at separation distance h.

Kriging when the Correlogram Is Known

Consider the model $Y(s)=m(s)+u(s)$. Here, m is the mean (which can vary spatially) and u represents the variation of Y about its mean, with $E[u]=0$. This model can be thought of as breaking the variation of Y into two components: large scale (m) and small, localized variation (u). In what follows, it is assumed that u is second order stationary, although the weaker intrinsic hypothesis can also be used.

The kriging predictors are unbiased in the sense that $E[P(s_0)]=E[Y(s_0)]$, where P is the predictor for Y at site s_0. Three types of kriging are discussed in the literature: simple, ordinary and universal. In simple kriging, m is assumed to be a known constant. In ordinary kriging, m is assumed to be an *unknown* constant. In universal kriging, m is unknown and varies spatially. In what follows, the kriging weights for the simple kriging case will be derived and the formulas for the other two cases will be introduced.

Simple Kriging[2]

Since m is assumed to be known in simple kriging, the focus is on predicting u. In kriging, the prediction is a weighted average of the data points:

$$\hat{u}(s_0) = \boldsymbol{\lambda}'\mathbf{u}\,, \tag{2}$$

where $\hat{u}(s_0)$ is the prediction at location s_0, $\mathbf{u}$ is a vector of the data, and $\boldsymbol{\lambda}$ is a vector of weights. $\boldsymbol{\lambda}$ is chosen by minimizing the expectation of the squared prediction errors. The expectation of the squared prediction errors can be written and simplified as shown below.

$$F = E\left[\left(u(s_0) - \hat{u}(s_0)\right)^2\right] \tag{3a}$$

$$= E\left[u(s_0)^2 - 2u(s_0)\hat{u}(s_0) + \hat{u}(s_0)^2\right] \tag{3b}$$

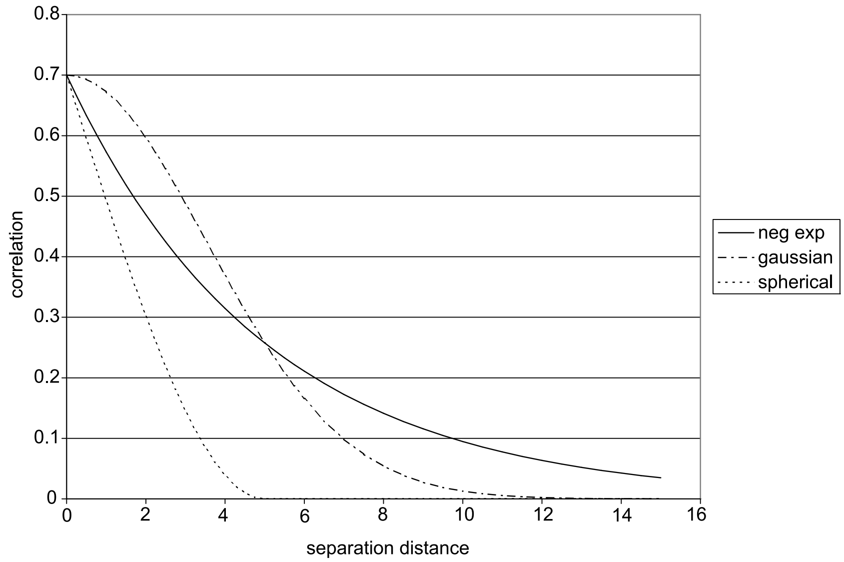
$$= \sigma^2 - 2E\left[u(s_0)\boldsymbol{\lambda}'\mathbf{u}\right] + E\left[\boldsymbol{\lambda}'\mathbf{u}\mathbf{u}'\boldsymbol{\lambda}\right] \tag{3c}$$

$$= \sigma^2 - 2\sigma^2\boldsymbol{\lambda}'\mathbf{k}(s_0) + \sigma^2\boldsymbol{\lambda}'\mathbf{K}\boldsymbol{\lambda}\,. \tag{3d}$$

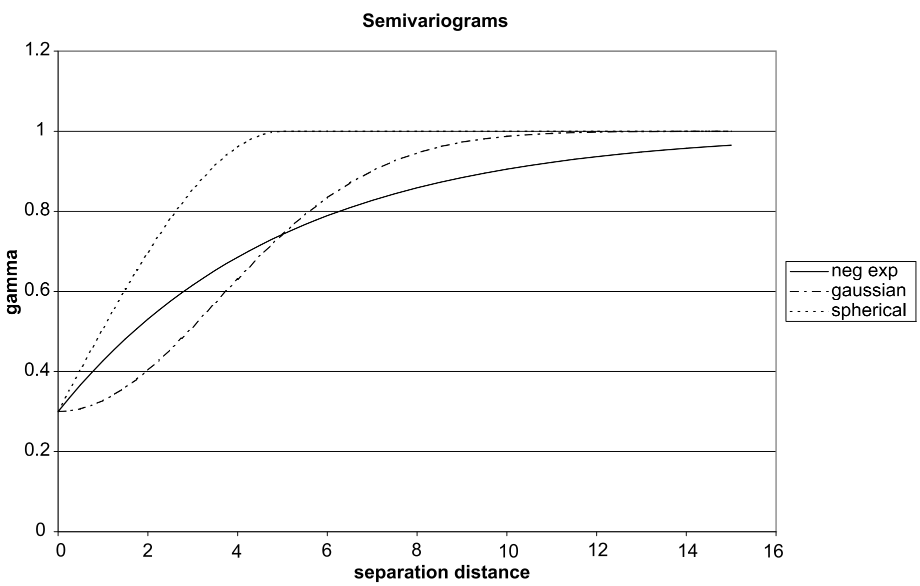
In this instance, $\mathbf{K}$ is the correlation matrix of the data (the correlations are calculated using the correlogram) and $\mathbf{k}(s_0)$ is a vector of correlations between location s_0 and

[1] In general, a location will be identified by a vector of coordinates. However, here s is used as a label rather than a vector of coordinates. Locations that are h units away from s are denoted by (s+h). Thus, both s and h are treated as scalars rather than vectors (the standard treatment in the literature is to treat them as vectors).

[2] This section draws heavily from Ripley, chapter 4.



Kriging, Figure 1 Three correlograms



Kriging, Figure 2 Three semivariograms

the data. Equation (3a) is the expected value of the squared prediction errors, where $u(s_0)$ is the true value of the surface at location s_0 (this value is unknown). In Eq. (3b), the term inside the brackets is squared. In Eq. (3c), the expectation operator is brought inside the bracket and (3d) simplifies the result under the assumption that $E[u]=0$. To find $\boldsymbol{\lambda}$ that minimizes F, take the derivative of F with respect to $\boldsymbol{\lambda}$, set it equal to zero, and solve for $\boldsymbol{\lambda}$:

$$\frac{\partial F}{\partial \boldsymbol{\lambda}} = -2\sigma^2 \mathbf{k}(s_0) + 2\sigma^2 \boldsymbol{\lambda} \mathbf{K} = 0 \tag{4}$$

$$\boldsymbol{\lambda} = \mathbf{K}^{-1} \mathbf{k}(s_0) \,. \tag{5}$$

Thus $\boldsymbol{\lambda}$ is a function of the correlations between the data points themselves and the correlations between the data points and the prediction point s_0. Equation 5 shows that data points which are closest to the prediction site will have higher weights since $\boldsymbol{k}(s_0)$ falls with separation distance. Also, data points which are in clusters will have less weight (this is due to the presence of $\mathbf{K}^{-1}$).

To predict Y at s_0, we simply add $\hat{u}$ to the known m: $\hat{Y}(s_0) = m + \hat{u}(s_0)$. This estimator is unbiased since $E[\hat{Y}(s_0)] = E[m + \boldsymbol{\lambda}'\mathbf{u}] = m = E[Y(s_0)]$.

A surface can be produced by predicting at locations lying along a reasonably fine grid. Kriging is a spatial interpolator: if s_0 is in the data set and there is no nugget effect, the kriging estimator will return the value of the datum at s_0. If there is a nugget effect, this will not be true and the surface at s_0 will be a weighted average of the surrounding data points (note that **K** must have ones on the main diagonal).

Ordinary Kriging[3] In ordinary kriging, the model is $Y(\mathrm{s}) = m + u(\mathrm{s})$; m is assumed to be constant over space but unknown. The kriging predictor is $\hat{Y}(s_0) = \boldsymbol{\lambda}'\mathbf{Y}$. Unbiasedness requires that

$$\begin{aligned} E\left[\boldsymbol{\lambda}'\mathbf{Y}\right] &= E\left[Y(s_0)\right] \\ &= E\left[\boldsymbol{\lambda}'(m + \mathbf{u})\right] = m \\ &= \boldsymbol{\lambda}'m = m\,. \end{aligned}$$

Thus, the prediction will be unbiased if $\sum_i \lambda_i = 1$. Therefore, in order to find $\boldsymbol{\lambda}$, minimize the expected prediction error subject to the constraint that $\sum_i \lambda_i = 1$. This can be accomplished by taking the derivatives of the Lagrangian with respect to $\boldsymbol{\lambda}$ and the Lagrange multiplier (μ).
If the weights are constrained to the sum to one, the formula for $\boldsymbol{\lambda}$ becomes

$$\boldsymbol{\lambda}' = \left(\mathbf{k}(s_0) + \mathbf{1}\frac{1 - \mathbf{1}'\mathbf{K}^{-1}\mathbf{k}(s_0)}{\mathbf{1}'\mathbf{K}^{-1}\mathbf{1}}\right)'\mathbf{K}^{-1}\,, \tag{6}$$

and the Lagrange multiplier is given by

$$\mu = \frac{\sigma^2\left(1 - \mathbf{1}'\mathbf{K}^{-1}\mathbf{k}(s_0)\right)}{\mathbf{1}'\mathbf{K}^{-1}\mathbf{1}}\,, \tag{7}$$

where $\mathbf{1}$ is a $1 \times n$ vector of ones. The only difference between Eq. 6 and Eq. 5 is the second term inside the parenthesis, which comes from the constraint.

Universal Kriging In universal kriging, the assumption that the mean is constant is dropped. The mean is generally hypothesized to be a low order polynomial function of the coordinates.[4] Let $\boldsymbol{x}(s)$ be a row vector of the polynomial terms and $\boldsymbol{\beta}$ be the associated coefficients. Then,

$$Y(s) = \mathbf{x}(s)\boldsymbol{\beta} + u(s)\,. \tag{8}$$

The kriging predictor is still $\hat{Y}(s_0) = \boldsymbol{\lambda}'\mathbf{Y}$. Unbiasedness requires that

$$\begin{aligned} E\left[Y(s_0)\right] &= E\left[\boldsymbol{\lambda}'\mathbf{Y}\right] \\ &= E\left[\mathbf{x}(s_0)\boldsymbol{\beta} + u(s_0)\right] = E\left[\boldsymbol{\lambda}'\left(\mathbf{X}\boldsymbol{\beta} + \mathbf{u}\right)\right] \\ &= \mathbf{x}(s_0)\boldsymbol{\beta} = \boldsymbol{\lambda}'\mathbf{X}\boldsymbol{\beta}\,, \end{aligned}$$

where the last line shows that the prediction will be unbiased if $\mathbf{x}(s_0) = \boldsymbol{\lambda}'\mathbf{X}$ since the β's cancel. Minimizing the expected prediction error subject to this constraint produces the following formula for $\boldsymbol{\lambda}$.

$$\begin{aligned} \boldsymbol{\lambda}' = \Big\{&\mathbf{k}(s_0) + \mathbf{X}\left(\mathbf{X}'\mathbf{K}^{-1}\mathbf{X}\right)^{-1} \\ &\cdot\left(\mathbf{x}(s_0)' - \mathbf{X}'\mathbf{K}^{-1}\mathbf{k}(s_0)\right)\Big\}'\mathbf{K}^{-1} \end{aligned} \tag{9}$$

Additionally, the Lagrange multipliers are

$$\boldsymbol{\mu}' = \sigma^2\left(\mathbf{x}(s_0)' - \mathbf{X}'\mathbf{K}^{-1}\mathbf{k}(s_0)\right)'\left(\mathbf{X}'\mathbf{K}^{-1}\mathbf{X}\right)^{-1}\,. \tag{10}$$

Again, the major difference between Eq. 9 and Eq. 5 is the second term inside of the brackets.

Kriging When the Correlogram is Unknown The foregoing discussion assumed that the covariance structure of **u** was known. However, it is far more common for the covariance structure to be unknown. In this case, it must be estimated. In order to make the estimation feasible, some assumptions regarding the covariance must be made. In particular, it is necessary to assume that the correlations follow some functional form, thereby reducing the estimation problem to estimating the parameters. Not all functions can serve as valid correlograms. In order for a correlogram function to be valid, it must be positive definite. The correlograms shown in Fig. 1 are all valid. The formulas are shown below.
Negative Exponential correlation function:

$$K(h) = \begin{cases} b_1 \exp\left(-\frac{h}{b_2}\right) & \text{for } h > 0 \\ 1 & \text{for } h = 0 \end{cases}\,. \tag{11}$$

Gaussian correlation function:

$$K(h) = \begin{cases} b_1 \exp\left(-\left(\frac{h}{b_2}\right)^2\right) & \text{for } h > 0 \\ 1 & \text{for } h = 0 \end{cases}\,. \tag{12}$$

Spherical correlation function:

$$K(h) = \begin{cases} b_1\left(1 - \frac{3h}{2b_2} + \frac{h^3}{2b_2^3}\right) & \text{for } 0 < h < b_2 \\ 1 & \text{for } h = 0 \\ 0 & \text{for } h > b_2 \end{cases}\,, \tag{13}$$

where $K(h)$ is the correlation of two observations separated by distance h, and b_1 and b_2 are the parameters to be estimated. Note that in Fig. 1, $b_1 = .7$ and $b_2 = 5$.
One approach for estimating the parameters of the correlation function is to form the empirical correlogram from the data. The parameters are estimated by fitting the chosen correlation function to the empirical correlogram (usually

[3] The material in this and the next section draw heavily on Cressie, chapter 3.

[4] In a model designed to predict housing prices, Dubin hypothesizes that the mean is a function of housing characteristics as well as the coordinates.

K

by eye). The empirical correlogram is formed by dividing the separation distances into ranges. Calculate the correlations for all the pairs falling into each distance range:

$$\hat{K}(h) = \frac{1}{N_h} \sum_{i,j \in N(h)} \left(Y(s_i) - \bar{Y}\right)\left(Y\left(s_j\right) - \bar{Y}\right), \tag{14}$$

where $N(h)$ is the set of all pairs falling within separation distance range h,

N_h is the number of distinct pairs in this set, and

$\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y(s_i)$ (i. e., the overall sample mean).

This approach only works if the mean is constant. If this is not the case, then $\hat{K}(h)$ will be contaminated by the changing mean and will produce poor estimates of the correlations.

If the mean cannot be assumed to be constant (as in universal kriging), then maximum likelihood can be used to simultaneously estimate $\boldsymbol{\beta}$ (the parameters of the mean) and $\boldsymbol{\theta}$ (the parameters of the correlogram). In this approach, we choose $\boldsymbol{\theta}$ to maximize the concentrated log likelihood function, L.

$$L = -\frac{N}{2} \times \ln\left(\left(\mathbf{Y} - \mathbf{X}\tilde{\boldsymbol{\beta}}\right)' \mathbf{K}(\boldsymbol{\theta})^{-1}\left(\mathbf{Y} - \mathbf{X}\tilde{\boldsymbol{\beta}}\right)\right) - \frac{1}{2}\ln|\mathbf{K}(\boldsymbol{\theta})| \,, \tag{15}$$

where $\tilde{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{K}(\boldsymbol{\theta})^{-1}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{K}(\boldsymbol{\theta})^{-1}\mathbf{Y}$.

However, Cressie notes that the ML estimators of $\boldsymbol{\theta}$ are negatively biased (that is, they tend to underestimate the spatial correlation in the data), particularly when the sample size is small. The bias comes from the dependence of L on both $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$. He suggests using the restricted maximum likelihood, in which the data is transformed so that the likelihood function depends only on $\boldsymbol{\theta}$ (Cressie (1991), p. 4).

Key Applications

Kriging has many applications. A few of these are listed below.

Cartography

Contour maps require interpolating from field measurements. Kriging can be used as a method of automatic contouring.

Mining

Kriging has been used to predict ore body locations from a series of sample cores.

Epidemiology

Kriging has been used to predict the incidence of various diseases such as SIDS.

Agriculture

Kriging has been used to predict crop yields and soil properties.

Network Design

Kriging can be used to determine the optimal deletion or addition of a network node.

Real Estate

Kriging can be used to predict house prices from housing characteristics.

Cross References

► Autocorrelation, Spatial
► Semivariogram Modeling
► Spatial Econometric Models, Prediction

Recommended Reading

1. Dubin, R.A.: Predicting House Prices Using Multiple Listings Data. Journal of Real Estate Finance and Economics **17**(1), 35–60 (1998)
2. Cressie, N.: Statistics for Spatial Data. Wiley and Sons, Inc., New York (1991)
3. Cressie, N., Read, T.R.C.: Do Sudden Infant Deaths Come in Clusters?. Statistics and Decisions, Supplement Issue **2**(3), 333–349 (1985)
4. Istok, J.D., Cooper, R.M.: Geostatistics Applied to Groundwater Pollution. Journal of Environmental Engineering **114**, 915–928 (1988)
5. Olea, R.: Optimum Mapping Techniques. Kansas Geological Survey, Lawrence, Kansas (1975)
6. Ord, J.K., Rees, M.: Spatial Processes: Recent Developments with Applications to Hydrology. In: Lloyd, E.H., et. al (eds.) The Mathematics of Hydrology and Water Resources. Academic Press, New York (1979)
7. Ripley, B.: Spatial Statistics. Wiley and Sons, Inc., New York (1981)
8. Samra, J.S., Gill, H.S., Bhatia, V.K.: Spatial Stochastic Modeling of Growth and Forest Resource Evaluation. Forest Science **35**, 663–676 (1989)

Land Administration System

► Cadastre

Land Cover Change Detection

► Change Detection

Land Information System

► Cadastre

Land Policy

► Cadastre

Land Registry

► Cadastre

Land Use Planning and Environmental Modeling

► Environmental Planning and Simulation Tools

Landmark Proximity

► Indoor Localization

Lane Reversal

► Contraflow in Transportation Network

Laser Altimetry (in Case of Airborne Platforms)

► Laser Scanning

Laser Scanning

GEORGE VOSSELMAN
ITC Department of Earth Observation Science, International Institute for Geo-Information Science and Earth Observation, Enschede, The Netherlands

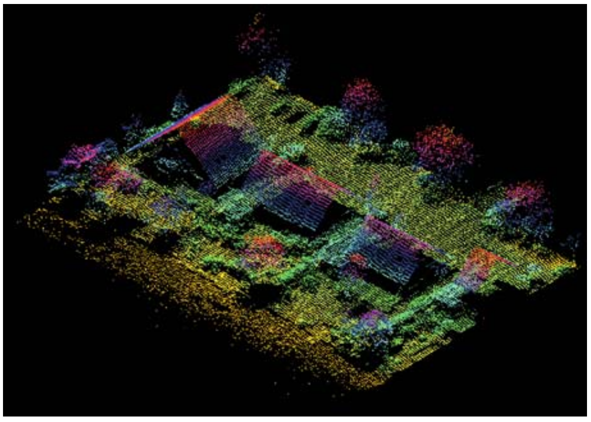
Synonyms

Laser altimetry (in case of airborne platforms); LiDAR; Light detection and ranging

Definition

Laser scanning is a surveying technology for rapid and detailed acquisition of a surface in terms of a set of points on that surface, a so-called point cloud. Laser scanners are operated from airborne platforms as well as from terrestrial platforms (usually tripods).

Operated from airborne platforms the pulses emitted by a laser range finder reflect on the Earth's surface and objects thereon and are again received by the laser range finder. The time elapsed between emitting and receiving a pulse is multiplied by the speed of light to derive the



Laser Scanning, Figure 1 Point cloud with a point density of 15 points/m^2

distance between the laser range finder and the reflecting object. This distance is combined with position and attitude information of the airborne platform, as well as the pointing direction of the laser beam, to calculate the location of the reflecting object. A rotating mirror is used to deflect the laser beam in directions perpendicular to the flight direction. Thus a swath below the flight path is scanned (Fig. 1).

Terrestrial laser scanners are usually operated from a stationary platform such as a tripod. In this case, the horizontal and vertical deflection of the laser beam is controlled by a system of two rotating mirrors. Various ranging techniques are applied for terrestrial laser scanning. Beside the "time-of-flight" principle, as applied by airborne laser scanners, continuous wave (phase measurements) and optical triangulation are used [1].

Historical Background

Airborne laser ranging was introduced in the 1960s shortly after the development of lasers. Seeking more effective techniques for topographic mapping of forested areas, laser range finders were combined with inertial navigation systems (INS) in the early 1980s [3]. A few years later the first experiments were conducted with the additional usage of global positioning system (GPS) receivers to improve the positional accuracy [4]. These experiments used only a downward pointing laser and thus only recorded height profiles below the flight path. While these laser profilers could only achieve a high point density in the across-flight direction at the cost of flying many flight lines, the experiments proved the high potential of ranging measurements combined with GPS and INS. In the early 1990s the first scanning laser range finders were introduced that allowed the acquisition of swaths. Terrestrial laser scanners were introduced to the market in the late 1990s.

Scientific Fundamentals

The laser range finders applied in laser scanners usually are equipped with infrared lasers. Typical wavelengths are 1,047, 1,064 and 1,540 nm. For bathymetry, green lasers are required (532 nm). Today's (2007) laser scanners fire up to 250,000 pulses of 4–10 ns/s and have ranging accuracies of 1–2 cm on well defined surfaces. The opening angles of the laser scanners vary in the range of 14–60°. As in aerial photography, large opening angles are used for surveys of open terrain, whereas small opening angles are preferred in urban areas to minimize occlusions. The rotation speed of the mirror generating the scanning pattern can be adapted such that the point spacing in flight direction is comparable to the point spacing within a scan line.

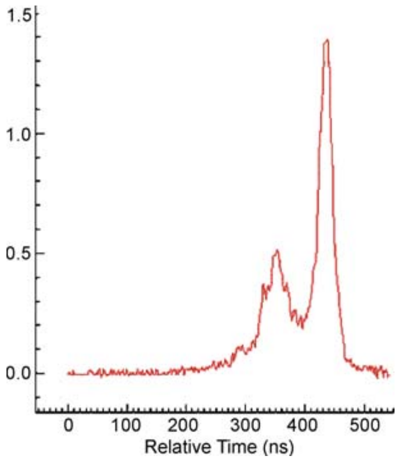
Position information of the airborne platform is obtained with kinematic differential GPS. The GPS positioning contributes the largest component to the error budget of a laser scanner. The accuracy varies with the distance between the GPS receiver on board and the GPS reference stations in the field as well as with the number of reference stations. Typical standard deviations are in the range of 5–10 cm for planimetry and 7–15 cm for height accuracy, although standard deviations of 2 and 3 cm respectively have been claimed by companies using a helicopter as platform and multiple reference stations.

The attitude information is obtained by an inertial measurement unit (IMU) consisting of accelerometers and gyroscopes. The standard deviations of 0.013° of the IMUs can be reduced to around 0.005° after post-processing together with the GPS observations.

The pulse frequency of the laser range finder is much higher than the measurement frequencies of the GPS (1–2 Hz) and IMU (200 Hz). Interpolation and accurate time registration of all observations is therefore required to obtain position and attitude information for each ranging observation. Another important aspect of the integration of these three measurement devices is the determination of their relative positions and orientations. On-site measurements in the aircraft are combined with calibrations during the post-processing phase to calculate the relationships between the coordinate systems of the devices.

For a beam divergence of 0.3 mrad the footprint of laser beams emitted at an altitude of 1,000 m are 30 cm in radius. Within this footprint multiple objects at various heights may reflect a part of the emitted light. Typical examples are footprints on vegetation and building edges. The signal returning at the laser range finder may therefore have a complex shape (Fig. 2). Most laser scanner systems are capable of storing up to four or five returned echoes, thus measuring four or five points for a single emitted pulse. The latest laser scanners also have the option to digitize and completely store the returned signal. These scanners are called full-waveform scanners [11].

The performance characteristics of terrestrial laser scanners primarily depend on the ranging technique applied. Most accurate are scanners using optical triangulation. While precision of 0.05 mm can be achieved, the maximum measurement range is restricted to a few meters. Continuous-wave scanners measure the phase of the outgoing and incoming amplitude modulated signal. The distance is obtained by multiplying the phase difference by the modulation wavelength. This leads to an ambiguous distance for objects further apart than one wavelength. The advantage of this scanner is their high measurement frequency of up to 600 kHz. Of all laser scanners time-of-flight scanners have the longest range. These scanners have



Laser Scanning, Figure 2 Waveform of a reflection from terrain and vegetation

a lower accuracy (10–20 mm) and measurement speed (< 30 kHz) but can measure distances up to about 1,000 m. For further information on terrestrial laser scanners the reader is referred to [1].

Key Applications

The major motivation for the development of airborne laser scanning was the acquisition of digital elevation models of forested areas. In the meantime, laser scanning has become the preferred technology for acquisition of digital elevation models for any type of terrain. Furthermore, the dense point clouds contain much information that can be used to extract other types of information.

Digital Elevation Models

Laser scanning enabled the acquisition of unprecedented high point density elevation models. These models are used for a wide variety of applications like water management (coastal zone management, hydrology, ground water level determination) and natural hazard studies (land slides, erosion, flooding). To extract a digital elevation model, all points reflecting from objects above the ground need to be removed from the raw laser scanner data. Various algorithms for this filtering operation have been developed [7].

Three-Dimensional City Models

High point density laser scanner data contains detailed information on the shapes of building roofs and can be used to produce 3D city models [2]. These models are used for urban planning, environmental studies (noise and gas propagation), as well as cellular phone network planning.

Forestry

In addition to information on the terrain height, laser scanner data of forested areas also contains information on the height and density of the vegetation [5]. This can be used for forest inventories as well as for biomass estimation. In particular, full-waveform scanners can provide much information on the density of canopies.

Corridor Mapping

Laser scanning has also proven useful for large-scale mapping around linear infrastructure such as roads, rail roads and power lines. In particular, when surveying from helicopters small objects like tracks and power lines can be automatically extracted from the laser scanner data.

Terrestrial

Applications of terrestrial laser scanning are mainly found in modeling industrial installations, scene reconstruction and preservation of cultural heritage. With high point densities of 1 point/cm^2, terrestrial laser scanning offers opportunities to create detailed surface models of facades as a part of high-resolution 3D city models.

Future Directions

The past years have shown rapid improvement in laser scanning technology. Pulse frequencies have increased, IMU technology has allowed a more accurate georeferencing and full-waveform scanners have been introduced. The potential of the resulting high-density information is still to be explored. Segmentation algorithms are being developed for automatic classification of laser scanner data as well as for 3D building modeling. It is expected that the processing of laser scanner data will be integrated with information extraction from optical sensors.

Recommended Reading

1. Fröhlich, C., Mettenleiter, M.: Terrestrial laser scanning—new perspectives in 3D surveying. Int. Arch. Photogramm. Remote Sensing Spatial Inf. Sci. **36**, 7–13 (2004). Available via http://www.isprs.org/commission8/workshop_laser_forest/FROEHLICH.pdf (2007). Accessed 6 July 2007
2. Haala, N., Brenner, C.: Virtual City Models from Laser Altimeter and 2D Map Data. Photogramm. Eng. Remote Sensing **65**, 787–795 (1999)
3. Krabill, W.B., Collins, J.G., Link, L.E., Swift, R.N., Butler, M.L.: Airborne laser topographic mapping results. Photogramm. Eng. Remote Sensing **50**, 685–694 (1984)
4. Lindenberger, J.: Test results of laser profiling for topographic terrain survey. In: Proceedings 42nd Photogrammetric Week, Stuttgart, 11–16 September 1989
5. Næsset, E., Gobakken, T., Holmgren, J., Hyyppä, H., Hyyppä, J., Maltamo, M., Nilsson, M., Olsson, H., Persson, Å., Söderman,

L

U.: Laser scanning of forest resources: the Nordic experience. Scand. J. Forest Res. **19**, 482–499 (2004)
6. Proceedings ISPRS Workshop "Laser scanning 2005", Int. Arch. Photogramm. Remote Sensing Spatial Inf. Sci. **36**, http://www.commission3.isprs.org/laserscanning2005/, Vosselman, G., Brenner, C. (Eds.), accessed July 7, 2007
7. Sithole, G., Vosselman, G.: Experimental comparison of filter algorithms for bare-earth extraction from airborne laser scanning point clouds. ISPRS J. Photogramm. Remote Sensing **59**, 85–101 (2004)
8. Theme issue "Airborne Laser Scanning": ISPRS J. Photogramm. Remote Sensing **54**
9. ISPRS Working Group III/3: Processing of point clouds from laser scanners and other sensors. http://www.isprs.org/commission3/wg3/
10. ISPRS Working Group V/3: Terrestrial laser scanning. http://www.isprs.org/commission5/wg3/
11. Wagner, W., Ullrich, A., Ducic, V., Melzer, T., Studnicka, N.: Gaussian decomposition and calibration of a novel small-footprint full-waveform digitising airborne laser scanner. ISPRS J. Photogramm. Remote Sensing **60**, 100–112 (2006)

Layer

▶ Spatial Data Transfer Standard (SDTS)

LBS

▶ Location-Aware Technologies

Legend

▶ Metadata and Interoperability, Geospatial

Level of Detail

▶ Hierarchies and Level of Detail

Library, Software

▶ Index Structures, Extensible

LiDAR

▶ Evolution of Earth Observation
▶ Laser Scanning

Life-Time

▶ Indexing Spatio-temporal Archives

Light Detection and Ranging

▶ Laser Scanning

Linear Constraint Databases

▶ Linear Versus Polynomial Constraint Databases

Linear Reference Model

▶ Road Network Data Model

Linear Versus Polynomial Constraint Databases

BART KUIJPERS
Theoretical Computer Science Group, Hasselt University and Transnational University of Limburg, Diepenbeek, Belgium

Synonyms

Linear constraint databases; First-order logic with constraints queries

Definition

The framework of *constraint databases* provides a rather general model for spatial databases [14]. In the constraint model, a spatial database contains a finite number of relations, that, although conceptually viewed as possibly infinite sets of points in the real space, are represented as a finite union of systems of polynomial equations and inequalities.

More specifically, in the *polynomial constraint database model*, a relation is defined as a boolean combination (union, intersection, complement) of subsets of some real space $\mathbb{R}^n$ (in applications, typically $n = 2$ or 3) that are definable by polynomial constraints of the form $p(x_1, \ldots, x_n) \geq 0$, where p is a polynomial in the real variables $x_1, \ldots, x_n$ with integer coefficients. For example, the spatial relation consisting of the set of points in the upper half of the unit disk in $\mathbb{R}^2$ can be represented by the formula $x^2 + y^2 \leq 1 \wedge y \geq 0$. In mathematical terminology, these spatial relations are known as *semi-algebraic* sets [4].

A basic query language for constraint databases is obtained by extending the relational calculus with polynomial inequalities [14]. This language is usually referred to as **FO + poly**. The query deciding whether the two-dimensional spatial relation S is a straight line, for instance, can be expressed in this language by the sentence

$$\exists a \exists b \exists c\big(\neg(a=0 \wedge b=0) \wedge \forall x \forall y\big(S(x,y) \leftrightarrow ax+by+c=0\big)\big)\,.$$

Formulas in the logic **FO + poly** can also be used to define output sets. For instance, the formula

$$\exists \varepsilon\big(\big(\varepsilon \neq 0 \wedge \forall x' \forall y'\big((x-x')^2+(y-y')^2 < \varepsilon^2 \rightarrow S(x',y')\big) \qquad (1)$$

defines the topological interior of a two-dimensional set S. Although variables in expressions such as (1) range over the real numbers, queries expressed in this calculus can still be computed effectively. In particular, the closure property holds: *any* **FO + poly** *query, when evaluated on a spatial database in the constraint model, yields a spatial database in the constraint model*. This follows immediately from Tarski's quantifier-elimination procedure for the first-order theory of real closed fields [17]. For instance, the evaluation of the query expressed by (1), on the input relation containing the upper half disk given by $x^2+y^2 \leq 1 \wedge y \geq 0$, can be obtained by plugging in this description of the input relation in the query formula where S appears, resulting in the formula

$$\exists \varepsilon\big(\varepsilon \neq 0 \wedge \forall x' \forall y'\big((x-x')^2+(y-y')^2 < \varepsilon^2 \rightarrow x'^2+y'^2 < q1 \wedge y' \geq 0\big)$$

and then eliminating the quantifiers. This would result in a formula like $(x^2+y^2=1 \wedge y \geq 0) \vee (y=0 \wedge -1 \leq x \leq 1)$, describing the output relation.

In the *linear constraint database model*, in the description of relations, only linear polynomial constraints are allowed and in the formulation of queries the relational calculus with linear inequalities is used. The latter language is usually referred to as **FO + lin**. Spatial relations definable by linear polynomial constraints are known as *semi-linear* sets. Also in the context of linear constraints there is a closure property: *any* **FO + lin** *query, when evaluated on a spatial database in the linear constraint model, yields a spatial database in the linear constraint model*.

Historical Background

The polynomial and linear constraint database models were introduced by Kanellakis, Kuper, and Revesz [11] in 1990. This model was well-studied in the 1990s and has become mainstream database research with practical and mathematical motivations. The research on the constraint database model has been published in all important database conferences and computer science journals. A state of the art book "Constraint databases," edited by G. Kuper, L. Libkin, J. Paredaens appeared in 2000 [12], and the textbook "Introduction to Constraint Databases" by P. Revesz was published in 2002 [15].

The PhD thesis of Luc Vandeurzen was devoted to the study of the linear constraint database model [18].

Scientific Fundamentals

Here, we discuss some differences in expressive power between **FO + poly** and **FO + lin**. In particular, we will illustrate that the expressive power of **FO + lin** is less than that of **FO + poly**.

As mentioned before, the topological interior of a two-dimensional set S can be expressed in **FO + poly** by the formula (1). Since the topology of $\mathbb{R}^2$ based on open discs is equivalent to the one based on open rectangles, we can equivalently express the topological interior of a semi-algebraic subset of $\mathbb{R}^2$ in **FO + lin** by the formula

$$\exists \varepsilon\big(\varepsilon > 0 \wedge \forall x' \forall y'\big((|x-x'| < \varepsilon \wedge |y-y'| < \varepsilon) \rightarrow S(x',y')\big).$$

But there are other queries for which the multiplication seems to be really necessary to express them. If we want to express that a two-dimensional semi-linear set is *convex*, for instance, then we can do this in **FO + poly** with the formula

$$\forall \vec{x} \forall \vec{y}\big(S(\vec{x}) \wedge S(\vec{y})\big) \rightarrow \forall \lambda\big(0 < q\lambda < q1 \rightarrow S(\lambda\vec{x}+(1-\lambda)\vec{y})\big).$$

Clearly, the subexpression $\lambda\vec{x} + (1-\lambda)\vec{y}$ uses quadratic polynomials and it may seem difficult to imagine that convexity of semi-linear sets might be expressible without multiplication. But it turns out that a semi-linear set of $\mathbb{R}^n$ is convex if and only if it is closed under taking midpoints. We can therefore express convexity of semi-linear sets by the **FO + lin**-formula

$$\forall \vec{x} \forall \vec{y}\big(S(\vec{x}) \wedge S(\vec{y})\big) \rightarrow \exists \vec{z}\big(2\vec{z} = \vec{x}+\vec{y} \wedge S(\vec{z})\big)\,.$$

We can wonder whether all **FO + poly** expressible queries and properties on semi-linear sets are also expressible in **FO + lin**. Clearly, **FO + poly** is more expressive than **FO + lin** for what concerns queries that return some n-dimensional result (the constant query that returns on any input the n-dimensional unit sphere, for instance, is not expressible in **FO + lin**). But also for what concerns properties of sets, it turns out that **FO + poly** is more expressive than **FO + lin**, as is illustrated by the following result: *the boolean query deciding whether a semi-linear subset S of $\mathbb{R}^2$ contains a line is expressible in* **FO + poly**, *but not in* **FO + lin** [3].

As another example, we give the boolean query deciding whether a semi-linear set S contains real numbers u and v satisfying $u^2+v^2=1$. This query is expressible in **FO + poly**, but not in **FO + lin** [1].

Several researchers have addressed the expressive power of **FO + lin** as a query language (see e. g., [9,18]). For **FO + lin**, also a number of more technical expressiveness results were obtained that concern the expressibility of on finite semi-linear sets. In particular, *generic* queries, i. e., queries that are invariant under certain transformations of the ambient space in which the semi-linear sets reside, are considered [2]. There are also a number of collapse results between different semantics of **FO + lin** (notably from natural semantics in which variables range over all real numbers, to active semantics, in which variables range over the active domain of the database). It was shown by Paredaens, Van den Bussche and Van Gucht that these semantics coincide [13].

Key Applications

Polynomial and linear constraint databases have their main applications in spatial databases and geographic information systems (GIS) [16]. In the vector model for GIS, data is often modeled by points, polylines and polygons. Points may represent objects, like buildings, trees or even cities (depending on the level of detail used). Polylines are often used to model streets and rivers and polygons to model regions like cities, provinces and states.

Clearly points, polylines and polygons are covered by the linear constraint model and this model therefore offers a lot of potential for applications in GIS. The constraint model has the advantage over other GIS database systems that it has well-defined logical query languages whose expressive power can be studied more precisely. It has the disadvantage that query evaluation depends on quantifier elimination, which has a too high complexity to be practically applicable.

The most important attempt to implement a GIS by means of linear constraints was undertaken at INRIA in Paris and resulted in the system DEDALE [6,7,8].

Linear constraint databases can also model applications that go beyond traditional GIS systems, in the sense that it can model time as an additional dimension. These applications typically require the representation of moving objects or spatiotemporal data. The PhD thesis of Sofie Haesevoets was devoted to the study of various constraint database models for spatio-temporal data [10].

Future Directions

Spatial constraint database systems that implement linear constraints have been developed. A prominent example of this is the system DEDALE system [6,7,8] that combines the use of a quantifier-elimination algorithm for the reals with linear constraints with techniques from computational geometry [5]. To achieve systems that are usable in the GIS practice, query evaluation algorithms will have to be improved. The high complexity of the quantifier-elimination algorithms, on which query evaluation relies, proves to be the bottleneck in developing usable linear constraint database systems.

Cross References

► Constraint Database Queries
► Constraint Databases and Data Interpolation
► Constraint Databases and Moving Objects
► Constraint Databases, Spatial
► Indexing Spatial Constraint Databases
► MLPQ Spatial Constraint Database System
► Visualization of Spatial Constraint Databases

Recommended Reading

1. Afrati, F., Cosmadakis, S., Grumbach, S., Kuper, G.: Linear versus polynomial constraints in database query languages. In: Borning, A. (ed.) Proceedings of the 2nd Workshop on Principles and Practice of Constraint Programming. Lecture Notes in Computer Science, vol. 874, pp. 181–192 Springer-Verlag, Heildelberg (1994)
2. Benedikt, M., Dong, G., Libkin, L., Wong, L.: Relational expressive power of constraint query languages. J. ACM **45**(1), 1–34 (1998)
3. Benedikt, M., Keisler, H.J.: Definability over linear constraints. In: Clote, P., Schwichtenberg, H. (eds.) Proceedings of Computer Science Logic, 14th Annual Conference of the EACSL. Lecture Notes in Computer Science, vol. 1862, pp. 217–231, Springer-Verlag, Heidelberg (2000)
4. Bochnak, J., Coste, M., Roy, M.-F.: Géométrie algébrique réelle. Springer-Verlag, Heidelberg (1987)
5. de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry: Algorithms and Applications. Springer-Verlag, Heidelberg (2000)
6. Grumbach, S.: Implementing linear constraint databases. In: Gaede, V., Brodsky, A., Günther, O., Srivastava, D., Vianu, V., Wallace, M. (eds.) Proceedings of the 2nd Workshop on Constraint Databases and Applications, Springer-Verlag, Berlin, New York, 1997
7. Grumbach, S., Rigaux, P., Scholl, M., Segoufin, L.: DEDALE, a spatial constraint database. In: Cluet, S., Hull, R. (eds.) Proceedings of the 6th International Workshop on Database Programming Languages (DBPL), Lecture Notes in Computer Science, vol. 1369, pp. 38–59. Springer, Berlin (1998)
8. Grumbach, S., Rigaux, P., Segoufin, L.: The DEDALE system for complex spatial queries. In: Proceedings of the 23th ACM International Conference on Management of Data (SIGMOD), pp. 213–224. ACM Press (1998)
9. Grumbach, S., Su, J., Tollu, C.: Linear constraint query languages: expressive power and complexity. In: Leivant, D. (ed.) Logic and Computational Complexity. Lecture Notes in Comput-

er Science, vol. 960, pp. 426–446, Springer-Verlag, Heidelberg (1995)
10. Haesevoets, S.: Modelling and Querying Spatio-temporal Data. PhD thesis, Limburgs Universitair Centrum (2005)
11. Kanellakis, P.C., Kuper, G., Revesz, P.Z.: Constraint query languages. J. Comp. Syst. Sci. **51**, 26–52 (1995)
12. Kuper, G.M., Libkin, L., Paredaens, J. (eds.) Constraint Databases. Springer-Verlag, Heidelberg (2000)
13. Paredaens, J., Van den Bussche, J., Van Gucht, D.: First-order queries on finite structures over the reals. SIAM J. Comp. **27**(6), 1747–1763 (1998)
14. Paredaens, J., Van den Bussche, J., Van Gucht, D.: Towards a theory of spatial database queries. In: Proceedings of the 13th ACM Symposium on Principles of Database Systems, pp. 279–288 (1994)
15. Revesz, R.Z.: Introduction to Constraint Databases. Springer-Verlag, Heidelberg (2002)
16. Rigaux, Ph., Scholl, M., Voisard, A.: Introduction to Spatial Databases: Applications to GIS. Morgan Kaufmann, (2000)
17. Tarski, A.: A Decision Method for Elementary Algebra and Geometry. University of California Press, Berkeley (1948)
18. Vandeurzen, L.: Logic-Based Query Languages for the Linear Constraint Database Model. PhD thesis, Limburgs Universitair Centrum (1999)

Linearization

► Indexing of Moving Objects, B^x-Tree
► Space-Filling Curves

Link-Node Model

► Road Network Data Model

LISA Statistics

► Local and Global Spatial Statistics

Local and Global Spatial Statistics

SUMEETA SRINIVASAN
School of Engineering and Applied Sciences,
Harvard University, Cambridge, MA, USA

Synonyms

Getis-Ord index G*; LISA statistics; Local Indicators of Spatial Association statistics; Local Moran's I; Statistics, spatial

Definition

These statistics assess the clustering of spatial data at the local level (using local clusters) or globally (using all the available data).

Main Text

Moran's I and Geary's C are global indices of spatial association that include all the locations in the data. A spatial contiguity matrix W_{ij}, with a zero diagonal, and the off-diagonal non-zero elements indicating contiguity of locations $\boldsymbol{i}$ and $\boldsymbol{j}$ are used to code proximities. The most commonly used global indicators of spatial autocorrelation are Moran's I and Geary's C which are defined as:

$$I = \frac{N \sum_i \sum_j W_{ij} Z_i Z_j}{\sum_i \sum_i W_{ij} \sum_i Z_i^2}\,, \tag{1}$$

$$C = \frac{(N-1) \sum_i \sum_j W_{ij}(x_i - x_j)^2}{2(\sum_i \sum_j W_{ij}) \sum_i Z_i^2}\,. \tag{2}$$

Z_i is the deviation of the variable of interest x_i from the mean $\bar{x}$ at location i, and N is the number of data points. Getis and Ord (1995) have defined several local measures including G^* which is defined as follows:

$$G_i^* * = \frac{\sum_j W_{ij}(d) x_j}{\sum_j x_j}\,. \tag{3}$$

Anselin (1995) defines a local version of the Moran's I as well as LISA statistics. These local measures have been used to identify spatial "hot-spots." The local Moran I_i is defined as:

$$I_i = Z_i \sum_j W_{ij} Z_j\,. \tag{4}$$

Local autocorrelation statistics make it possible to assess the spatial association of a variable within a particular distance of each observation.

Cross References

► Spatial Contiguity Matrices

Recommended Reading

Anselin, L.: Local indicators of spatial association – LISA. Geogr. Anal. **27**, 93–115 (1995)
Ord, J.K., Getis, A.: Local Spatial Autocorrelation Statistics: Distributional Issues and an Application. Geogr. Anal. **27**, 286–306 (1995)
Cliff, A., Ord, J.K.: Spatial Autocorrelation. Pion, London (1973)

Local Indicators of Spatial Association Statistics

► Local and Global Spatial Statistics

Local Moran's I

► Local and Global Spatial Statistics

L

Local Sensitivity Analysis

XIAOBO ZHOU, HENRY LIN
Department of Crop and Soil Sciences, The Pennsylvania State University, University Park, PA, USA

Definition

Local sensitivity analysis is the assessment of the local impact of input factors' variation on model response by concentrating on the sensitivity in vicinity of a set of factor values. Such sensitivity is often evaluated through gradients or partial derivatives of the output functions at these factor values, i. e., the values of other input factors are kept constant when studying the local sensitivity of an input factor.

Main Text

In large and complex models, it is often the case that the importance of input factors is not clear. In such cases, it is good to know the slope of the model's response at a set of given points in the factor space corresponding to a small change around these points. The investigated points where slopes are estimated are called nominal values of factors, and usually are the points of the best factor estimate. The simplest way to calculate local sensitivity is the so-called brute-force method, which requires the model to be completely recomputed to test the sensitivity to each individual parameter over a defined range. However, this method is slow and requires a trial-and-error test to determine the factor perturbations. Other local sensitivity analysis methods include the direct method, the Green's function method, the forwarded sensitivity analysis procedure (FSAP), and the adjoint sensitivity analysis procedure (ASAP). The local sensitivity analysis is not appropriate when the model is nonlinear and the uncertainties of input factors are in different orders of magnitude.

Cross References

► Global Sensitivity Analysis
► Screening Method
► Sensitivity Analysis

Locality-Preserving Mapping

► Space-Filling Curves

Localization

► Indoor Positioning, Bayesian Methods

Localization, Cooperative

NEAL PATWARI
Department of Electrical and Computer Engineering, University of Utah, Salt Lake City, UT, USA

Synonyms

Relative location; Distributed localization; Ad-hoc localization

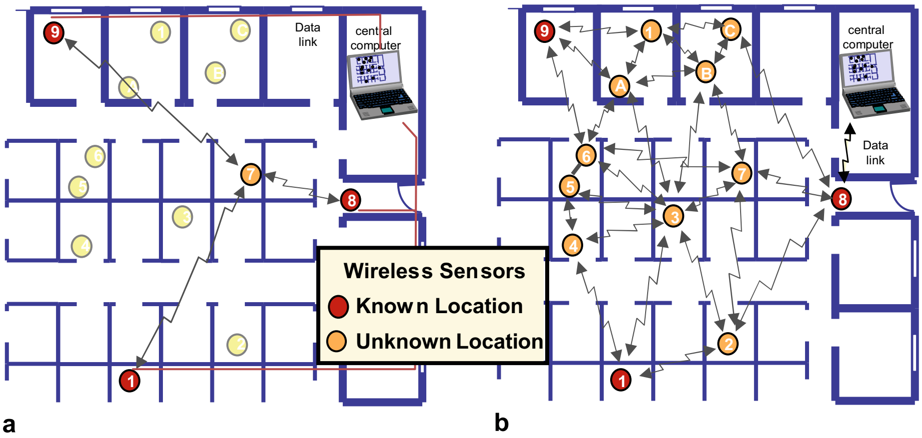
Definition

Cooperative Localization is the estimation of the locations of wireless devices (a.k.a. nodes) in a network using measurements made between many pairs (or subsets) of the nodes. While many localization methods limit an unknown-location device to making measurements with known-location nodes, cooperative localization methods specifically encourage measurements to be made between nodes regardless of each node's prior location knowledge. Then, cooperative localization algorithms use the 'mesh' of measurements to simultaneously estimate the coordinates (or relative coordinates) of all nodes.

Historical Background

Estimating the coordinates of a point in space based on measurements to other points in space which have only imperfect coordinate estimates themselves has a long history, notably in surveying, (see chapter on *Landmarks for Wayfinding*). Recently, interest in cooperative localization has stemmed from the research and development of large networks of low-cost, low-energy wireless sensors. Hundreds or thousands or more sensors will be deployed in a single environmental monitoring, precision agriculture, or indoor building monitoring and control system. To keep costs low in large systems, device costs must be kept low. GPS is seen as too expensive and too energy-intensive to be included on such simple devices, and early published methods emphasized that they were 'GPS-less' or 'GPS-free' [1,2].

Infrastructure costs can be high if base stations must be deployed as part of a localization system, as was true in *local-positioning systems* (LPS) which operated analogous to the global positioning system (GPS), but with transmitters installed on Earth to cover a local area. Sensor networks, which promise to require so little installation that they can be dropped out of a helicopter, avoid using significant fixed infrastructure. Cooperative localization requires little infrastructure, since few nodes need to be 'base stations' (have known-location) and low power nodes can be located even when they are in range of no known-location



Localization, Cooperative, Figure 1 Traditional multilateration or multi-angulation (**a**) uses only measurements made between an unknown-location node and multiple known-location nodes. Cooperative localization (**b**) additionally allows measurements between any pairs of nodes to aid in the location estimate

nodes, as long as they are within range of any other nodes. Early published research refers to the 'ad-hoc' [3] nature of cooperative localization.

The use of the word 'cooperative' originated with Savarese, Rabaey, and Beutel [4], who coined 'cooperative ranging' to describe using the high connectivity of a network to make measurements and perform local optimizations in the effort to find global position solutions across the network.

Scientific Fundamentals

Formal Problem Description

Formally, the cooperative localization problem displayed in Figs. 1 and 2 is the estimation of coordinates in a N node wireless network. In general, N unknown-location node D-dimensional coordinates must be estimated, depending on the prior coordinate knowledge which exists for the nodes:

- **No prior coordinate knowledge exists**: Only relative coordinates can be calculated (which have an arbitrary translation and rotation), and only the relative position of nodes is important;
- **Some imperfect prior coordinate knowledge exists**: If possible, we may improve upon the prior coordinate knowledge of those nodes with only imperfect prior knowledge; or
- **Coordinates of m nodes are perfectly known**: In this case, these m coordinates are accepted as fact, and only $n = N - m$ coordinates must be estimated.

Nodes with either of the last two types of prior information are typically referred to as 'known-location', 'reference', 'beacon', or 'anchor' nodes.

Measurements between nodes are the means to improve upon any existing prior information. These measurements are typically pairwise, that is, two nodes participate in each measurement, and $X_{i,j}$ is used to denote the measurement made between nodes i and j. Cooperative localization is unique in that node i and node j may both have no prior coordinate information, yet $X_{i,j}$ is still used in the coordinate estimation algorithm. Cooperative localization systems may also use *three-wise* or *four-wise* measurements; they are not limited to pairwise measurements. These many measurement methods are described in the next section.

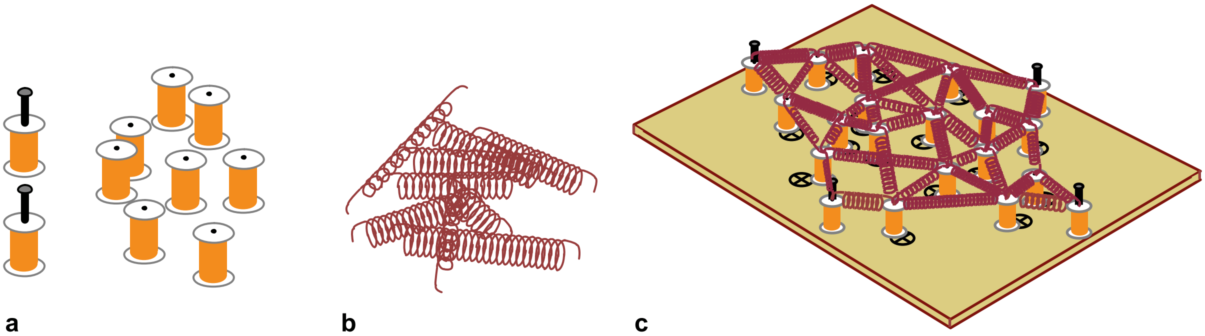
Location estimation is often inextricably tied with other *self-calibration* problems in wireless networks [5]. Time-synchronization, node angular orientation estimation, and transmit power level calibration, are all important calibration problems in wireless networks, and are discussed in the following section.

Methods of Distance Measurement

In an ideal, free-space environment, distance could be precisely estimated from any one of the following measurement methods. However, in a real-world environment full of obstructions, a measurement can only provide some imperfect information. As discussed in the chapter on *Channel Modeling and Algorithms for Indoor Positioning*, the key theme of the real-world environments is that multipath, that is the multiple signal copies of the transmission which arrive at the receiver from different angles with different phase shifts, time delays, and attenuations, cause ambiguity in the relative location of the transmitter with respect to the receiver. Here, we discuss some of the distance measurements particularly suited for cooperative localization, and their properties and sources of error.

Received Signal Strength

Measurements of field strength at a receiver are referred, often ambiguously, using the following terms:



Localization, Cooperative, Figure 2 Cooperative localization from pairwise distance measurements is analogous to finding the resting point of (**a**) masses (spools of thread) connected by a network of (**b**) springs. First, nodes with known-location are nailed to their known coordinates on a board. Springs have a natural length equal to measured ranges and can be compressed or stretched. They are connected to the pair of masses whose measured range they represent. After letting go, the equilibrium point (**c**) of the masses represent a minimum-energy localization estimate; the actual node locations are indicated by ⊗

- *Received signal strength (RSS)*: The voltage of the signal impinging on the receiver antenna.
- *Received signal strength indicator (RSSI)*: Specifically refers to the number calculated by the receiver's measurement circuitry, which is a measurement of the RSS.
- *Received Power*: The squared magnitude of the RSS.

These quantities are usually measured on a decibel (dB) scale. In dB, they are generally proportional to the logarithm of the distance between the transmitter and receiver. The ensemble mean received power at distance d, $\bar{P}(d)$, is typically modeled as

$$\bar{P}(d) = P_0 - 10 n_p \log \frac{d}{d_0} \tag{1}$$

where P_0 is the received power in dBm (decibel milliwatt) at a short reference distance d_0, and n_p is the 'path-loss exponent'. In free space, $n_p = 2$, but in real-world environments, n_p can be range from below 2 up to 5 for typical propagation environments.

Major Sources of Error For a particular measured power, $P_{i,j}$, the dBm power measured at receiver i of the power transmitted by node j, the multipath channel causes variation about the ensemble mean. Received power is a function of frequency because of frequency selective fading. Position dependent variation in RSS measurements are caused by shadowing, that is, the attenuation of a signal due to obstructions (furniture, walls, trees, buildings, and more) that a signal must pass through or diffract around in its path between the transmitter and receiver. Motion in the environment (movement of leaves in the wind, or of people or vehicles) causes temporal variation in received power. To reduce statistical variations, RSS measurements are typically averaged over time and frequency. However, for stationary nodes, significant errors can remain due to shadowing.

Statistical Model The difference between a measured received power and its ensemble average, due to shadowing, is modeled as log-normal (that is, Gaussian if expressed in dB). The log-normal model is based on a wide variety of measurement results and analytical evidence [6,7]. The standard deviation of received power (when received power is expressed in dBm), σ_{dB}, has units of (dB) and is relatively constant with distance. Typically, σ_{dB} is as low as 4 and as high as 12 [6]. Thus, the received power (dBm) at node i transmitted by j, $P_{i,j}$ is distributed as Gaussian with mean $\bar{P}(d_{i,j})$ and variance σ^2_{dB}, where the actual transmitter-receiver separation distance $d_{\mathrm{i,j}}$ is given by

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\,. \tag{2}$$

RSS-based range estimates have variance proportional to their actual range. Constant standard deviation in dB results in multiplicative factors which are constant with range. RSS errors are referred to as multiplicative, in comparison to the additive TOA errors presented in the Section "Time-of-Arrival" below. RSS is thus most valuable in high-density sensor networks.

Calibration and Synchronization Measured RSS is also a function of the transmitter and receiver. RSSI circuits and transmit powers will vary from device to device. Also, transmit powers decrease as batteries deplete. Typically, knowledge of the path loss exponent parameter, n_p, is assumed known, but it might also be estimated simultaneously with location [8].

To reduce synchronization requirements, *RSS difference*, measured at a pair of receivers to indicate their relative distance from the transmitter, can be used as the measurement statistic [9]. Other non-parametric approaches to RSS-based localization use only the order statistics of the RSS values measured at a receiver [10] to calculate cooperative location. Analysis has shown that cooperative localization systems can tolerate uncalibrated transmitters, but that both $P_{i,j}$ and $P_{j,i}$, not just their average, must be used by the estimator [11].

Time-of-Arrival

Time-of-arrival (TOA) methods estimate the clock time at the receiver of the arrival of the signal that traveled in the straight line from the transmitter to receiver (that is the line-of-sight or LOS path). The time difference between transmission and reception times is the product of the distance and the propagation speed.

Sources of Error Generally errors in TOA estimation are caused by two problems:

- *Early-Arriving Multipath*: Many multipath signals arrive very soon after the LOS, and their contributions severely obscure its estimation. Essentially, the early-arriving multipath serve as self-interference that effectively decreases the SNR of the desired LOS signal.
- *Attenuated LOS*: The LOS signal can be severely attenuated due to environmental obstructions in the path between the transmitter and receiver. When the LOS signal is too low compared to the early-arriving multipath components, it can be 'lost in the noise' and missed completely, causing large positive errors in the TOA estimate.

Measuring TOA between nearby neighbors is an advantage of dense networks. As the path length decreases, the LOS signal power (relative to the power in the multipath components) generally increases [12]. Thus, the severely attenuated LOS problem is only severe in networks with large inter-device distances. Generally, wider signal bandwidths are necessary for obtaining greater temporal resolution. Wideband direct-sequence spread-spectrum (DS-SS) or ultra-wideband (UWB) signals are popular techniques for high-bandwidth TOA measurements [13].

Statistical Model Measurements have shown that for short-range measurements, measured time delay can be roughly modeled as Gaussian, with mean $d_{i,j} / v_p + \mu_T$ and variance $\sigma_T{}^2$, where μ_T and $\sigma_T{}^2$ are the mean and variance of the time delay error, $d_{i,j}$ is given in (2), and v_p is the propagation velocity.

The presence of large errors (also called non-line-of-sight (NLOS) errors) can complicate the Gaussian model. These errors make the tails of the distribution of measured TOA heavier than Gaussian, and have been modeled using a mixture distribution: with a small probability, the TOA measurement results from a different, higher-variance distribution, as described in [14,15]. For TOA measurements made over time in a changing channel, the TOAs which include excess delays can be identified and ignored [14]. Even in static channels, if the number of range measurements to a device are greater than the minimum required, the redundancy can be used to identify likely NLOS errors [16].

Calibration and Synchronization In general, time synchronization of the clocks at nodes i and j would be required to relate a single TOA measurement between the nodes to the distance between them. For RF signals, each *ns* of clock error results in 1 foot (30.5 cm) of distance error, yet ns-level synchronization is impractical for most simple wireless devices. Other TOA measurements use acoustic (or ultrasound) or a combination of acoustic and RF signalling. For acoustic signals, only millisecond-accuracy synchronization is required. RF can be used for synchronization while acoustic is used for TOA measurement. RF-only systems can be used when performing *two-way* (or *round-trip*) TOA measurements [17,18]. In another method, one-way TOA measurements are used in an algorithm which simultaneously estimates sensor coordinates and clock synchronization [19].

Other Measurement Modalities

Radio Interferometric Measurements Radio interferometry is a four-wise (as opposed to pair-wise) cooperative measurement method. This method requires two receivers to measure the beat signal caused by the interference of two transmitters' continuous wave (CW) signals, as introduced in [20]. The difference in phase between the two received beat signals is a function of the relative positions of the four nodes. Measurements made using many different subsets of four nodes can be made. Coordinate optimization generally requires significant centralized computation, but results indicate the possibility of distributed estimation [21].

Angle-of-Arrival Measurements of the angle from which an RF signal arrives at the receiver are called angle-of-arrival (AOA) measurements, and require an antenna array (a series of antenna elements and RF chains) at the receiver. An antenna array's size is proportional to the carrier wavelength. Multipath arrive from a wide range

L

of angles, and the receiver must try to estimate the signal arriving from the direct path. Research has proposed to simplify AOA using the *RSS ratio method*, using only two elements with differing beam patterns to measure the ratio of the two elements' RSS to estimate AOA [22]. Such a method has reported a standard deviation of angular error of $\sigma_\alpha = 3°$. Measurements of AOA are only meaningful when sensor orientation is known, so calibration must include orientation estimation.

Node Mobility Node mobility can be exploited to increase the quantity of measurements. Systems may use GPS-capable mobile nodes, or tracked mobile sources, to achieve cooperative localization gains [23,24,25,26].

Accuracy

The accuracy of cooperative localization depends greatly on system implementation. Generally, the more accurate the measurement method, the more accurately a coordinate can be measured. Also, the more neighbors that each node has, the more robust and refined a coordinate estimate that it can obtain. For example, in a d-dimensional cooperative localization system with N nodes, each node making pairwise measurements with k neighbors, there are kN equations (obtained from the kN measurements) and less than dN unknowns. When k is much significantly higher than d, the extra constraints can be used to reduce the effect of the measurement errors, and even discard measurements that seem contradictory.

There are means to analytically lower bound both the variance and mean-squared error (MSE) of cooperative localization estimators [7,11,13,19,27,28,29,30]. These methods input the geometry, the type and quality of the measurements, the location and any prior information of the known-location nodes, and any unknown or random nuisance parameters, and then output a lower bound on the covariance or mean-squared error possible in the cooperative localization system. Such bounds allow system designers to select measurement type, network geometry and density, and prior information, to meet system specifications. Such bounds don't require simulation and thus can be calculated prior to investigating algorithms. Further, they provide algorithm developers a stopping point, since once an algorithm performs as well as or close to the lower bound, there isn't much point in further algorithm refinement.

Algorithms

A wide variety of algorithms have been proposed and evaluated for use in cooperative localization systems. Algorithms can be divided into *centralized algorithms*, which collect measurements to a central location (fusion center) for processing, and distributed, which sensors participate in a decentralized processing algorithm. For applications in which a higher-capability device, that is, laptop or PDA (personal digital assistant) is used to display information to a user, a centralized algorithm may be acceptable. However, as N increases, the communication costs of sending information to a central processor will become higher than the cost of a distributed algorithm [31].

Distributed algorithms for cooperative localization generally fall into one of two categories:

1. *Network Multilateration*: Each sensor estimates its multi-hop range to the nearest reference nodes (see *Fastest-path Computation*). These ranges can be estimated via the shortest path between the sensor and reference nodes, that is, proportional to the number of hops, or the sum of measured ranges along the shortest path [3,32]. Note that finding the shortest path is readily distributed across the network. When each sensor has multiple range estimates to known positions, its coordinates are calculated locally via multi-lateration.
2. *Successive Refinement*: These algorithms try to find the optimum of a global cost function, for example, least squares (LS), weighted LS [33], or maximum likelihood (ML). Each sensor estimates its location and then transmits that assertion to its neighbors [4,34,35]. Neighbors must then recalculate their location and transmit again, until convergence. The local calculation is typically minimal – one hardware implementation on the Texas Instruments CC2431 chip calculates the ML solution in 40 μs [36]. A device starting without any coordinates can begin with its own local coordinate system and later merge it with neighboring coordinate systems [2]. Typically, better statistical performance is achieved by successive refinement compared to network multilateration, but convergence issues must be addressed.

Bayesian networks (or factor graphs, or particle filters) provide a distributed successive refinement method to estimate the probability density function of node coordinates (see the chapters on *Bayesian Indoor Positioning Methods* and *Bayesian Networks*). In these methods, each sensor stores a conditional density on its own coordinates, based on its measurements and the conditional density of its neighbors [25,37].

Key Applications

Cooperative localization has been proposed for indoor and outdoor applications; civilian and military applications;

those requiring centimeter accuracy, and those requiring only localization within a few tens of meters.

Environmental Monitoring

In precision agriculture or environmental monitoring applications, when using cooperative localization, only a small number of sensors would need to be manually located or include a GPS receiver. The rest could be dropped into the soil from a tractor or helicopter and then automatically located. Further, recalculation of sensor locations will be possible when re-tilling soil, or if sensors move due to wind or precipitation. Cooperative localization animal tags can record the tracks and relative locations of tagged animals in a pack or group over an extended period of time. When relative animal positions are more important than absolute positions, GPS can be completely avoided, reducing energy consumption.

Logistics

Short-range active radios are proposed for 'bar-code replacement' applications. While radio-frequency identification (RFID) tags (see the chapter on *Radio Frequency Identification*) can be located only when they are in range of a RFID reader, tags in a cooperative localization system could be located as long as they are in range of other cooperative tags. Thus cooperative localization systems fill a important logistics need of warehouses and manufacturing floors – being able to accurately locate inventory or parts quickly and at any time.

Robust Localization

Cooperative localization may complement GPS in other wireless networks. Groups who communicate by radio, for example military units or emergency workers deployed to a critical location (for example, burning building), can be located by GPS as a primary source of location information. However, since GPS signals are too weak indoors, and can be jammed or intentionally scrambled, cooperative localization provides secondary location awareness, allowing workers to be located in the building, and allowing groups to know their relative location with respect to others in their group. Emergency localization of cellular phones may use cooperative techniques using secondary networks such as Bluetooth or WiFi.

Geographic-Based Routing

Message routing, as discussed in the chapter *Routing Algorithms*, is a critical issue in large wireless networks. If each node knows its coordinate, coordinate-based or geographic-based routing algorithms can reduce the energy and memory complexity at each node. Cooperative localization techniques enable these complexity-reducing routing methods.

Future Directions

A variety of research topics in cooperative localization remain important areas of future research:

- *Mobile Cooperative Localization and Tracking*: Tracking mobile nodes; reducing algorithm latency; using measurements over time and space to reduce errors; and distributed *change detection*.
- *Localization Security and Privacy*: Preventing and identifying attacks on the distributed measurement process, data storage, or algorithm, as in [38].
- *Multi-Modal Approaches*: Combining multiple measurement methods to achieve higher robustness and accuracy.
- *3-D Accuracy*: Development of systems with highest accuracy in the vertical dimension.
- *Opportunistic Approaches*: Using ambient, spatially correlated signals in the environment to estimate relative node positions, as in [29].

L

Cross References

► Channel Modeling and Algorithms for Indoor Positioning
► Geosensor Networks
► Geosensor Networks, Formal Foundations
► Indoor Localization
► Indoor Positioning, Bayesian Methods
► Routing Vehicles, Algorithms

Recommended Reading

1. Bulusu, N., Heidemann, J., Estrin, D.: GPS-less low cost outdoor localization for very small devices. IEEE Personal Comm. **5**(5), 28–34 (2000)
2. Čapkun, S., Hamdi, M., Hubaux, J.P.: GPS-free positioning in mobile ad-hoc networks. In: 34th IEEE Hawaii Int. Conf. on System Sciences (HICSS-34), Jan. (2001)
3. Niculescu, D., Nath, B.: Ad hoc positioning system. In: IEEE Globecom 2001, vol. 5, pp. 2926–2931 (2001)
4. Savarese, C., Rabaey, J.M., Beutel, J.: Locationing in distributed ad-hoc wireless sensor networks. In: ICASSP, May 2001, pp. 2037–2040 (2001)
5. Whitehouse, K., Culler, D.: Calibration as parameter estimation in sensor networks. In: WSNA '02: Proc. ACM Int. workshop on Wireless sensor networks and applications. (2002)
6. Rappaport, T.S.: Wireless Communications: Principles and Practice. Prentice-Hall Inc., New Jersey (1996)
7. Patwari, N., Hero III, A.O., Perkins, M., Correal, N., O'Dea, R.J.: Relative location estimation in wireless sensor networks. IEEE Trans. Signal Process. **51**(8), 2137–2148 (2003)

8. Li, X: RSS-based location estimation with unknown pathloss model. IEEE Trans. Wireless Commun. **5**(12), 3626–3633 (2006)
9. Liu, C., Wu, K., He, T.: Sensor localization with ring overlapping based on comparison of received signal strength indicator. In: IEEE Mobile Ad-hoc and Sensor Systems (MASS), Oct. (2004)
10. Yedavalli, K., Krishnamachari, B., Ravula, S., Srinivasan, B.: Ecolocation: A sequence based technique for RF-only localization in wireless sensor networks. In: Proc. 4th Int. Conf. on Information Processing in Sensor Networks (IPSN '05), April (2005)
11. Patwari, N., Hero III, A.O.: Signal strength localization bounds in ad hoc & sensor networks when transmit powers are random. In: Fourth IEEE Workshop on Sensor Array and Multichannel Processing (SAM'06), July (2006)
12. Pahlavan, K., Krishnamurthy, P., Beneat, J.: Wideband radio propagation modeling for indoor geolocation applications. IEEE Comm. Magazine **36**, 60–65 (1998)
13. Patwari, N., Ash, J., Kyperountas, S., Moses, R.M., Hero III, A.O., Correal, N.S.: Locating the nodes: Cooperative localization in wireless sensor networks. IEEE Signal Process. Mag. **22**(4), 54–69 (2005)
14. Borràs, J., Hatrack, P., Mandayam, N.B.: Decision theoretic framework for NLOS identification. In: IEEE VTC 1998, **2**, 1583–1587 (1998)
15. Gustafsson, F., Gunnarsson, F.: Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements. IEEE Signal Processing Mag., **22**(4), 41–53 (2005)
16. Ash, J.N., Moses, R.L.: Outlier compensation in sensor network self-localization via the EM algorithm. In: IEEE ICASSP, March 19–23, (2005)
17. Kim, S., Pals, T., Iltis, R.A., Lee, H.: CDMA multipath channel estimation using generalized successive interference cancellation algorithm for radiolocation. In: 36th Conf. Information Sciences & Systems, March (2002)
18. McCrady, D.D., Doyle, L., Forstrom, H., Dempsy, T., Martorana, M.: Mobile ranging with low accuracy clocks. IEEE Trans. Microwave Theory Tech. **48**(6), 951–957 (2000)
19. Larsson, E.G.: Cramér-Rao bound analysis of distributed positioning in sensor networks. IEEE Signal Process. Lett. **11**(3), 334–337 (2004)
20. Maroti, M., Kusy, B., Balogh, G., Volgyesi, P., Molnar, K., Nadas, A., Dora, S., Ledeczi, A.: Radio interferometric positioning. Institute for Software Integrated Systems, Vanderbilt University, Tech. Rep. ISIS-05–602, Nov. (2005)
21. Patwari, N., Hero III, A.O.: Indirect radio interferometric localization via pairwise distances. In: Third IEEE Workshop on Embedded Networked Sensors (EmNets'06), May (2006), pp. 26–30
22. Ash, J.N., Potter, L.C.: Sensor network localization via received signal strength measurements with directional antennas. In: Proceedings of the 2004 Allerton Conference on Communication, Control, and Computing (2004)
23. Cehver, V., McClellan, J.H.: Sensor array calibration via tracking with the extended Kalman filter. In: Proceedings of 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 5, pp. 2817–2820 (2001)
24. Galstyan, A., Krishnamachari, B., Lerman, K., Pattem, S.: Distributed online localization in sensor networks using a moving target. In: IEEE Information Processing in Sensor Networks (IPSN), April 2004, pp. 61–70 (2004)
25. Hu, L., Evans, D.: Localization for mobile sensor networks. In: MobiCom '04: Proc. 10th Int. Conf. on Mobile computing and networking, pp. 45–57 (2004)
26. Pathirana, P., Bulusu, N., Jha, S., Savkin, A.: Node localization using mobile robots in delay-tolerant sensor networks. IEEE Trans. Mob. Comp. **4**(4), 285–296 (2005)
27. Catovic, A., Sahinoglu, Z.: The Cramér-Rao bounds of hybrid TOA/RSS and TDOA/RSS location estimation schemes. Mitsubishi Electric Research Lab, Tech. Rep. TR-2003-143, Jan. 2004. [Online]. http://www.merl.com/reports/docs/TR2003-143.pdf
28. Moses, R.L., Krishnamurthy, D., Patterson, R.: A self-localization method for wireless sensor networks. EURASIP J. App. Sig. Proc. **4**, 348–358 (2003)
29. Patwari, N., Hero III, A.O.: Manifold learning algorithms for localization in wireless sensor networks. In: IEEE Intl. Conf. on Acoustic, Speech, & Signal Processing (ICASSP'04). **3**, 857–860 (2004)
30. Savvides, A., Garber, W., Adlakha, S., Moses, R., Srivastava, M.B.: On the error characteristics of multihop node localization in ad-hoc sensor netwoks. In: 2nd Intl. Workshop on Inform. Proc. in Sensor Networks, April 2003
31. Rabbat, M., Nowak, R.: Distributed optimization in sensor networks. In: IPSN '04: Proc. 3rd Int. Symp. on Information processing in sensor networks, pp. 20–27 (2004)
32. Savvides, A., Park, H., Srivastava, M.B.: The bits and flops of the n-hop multilateration primitive for node localization problems. In: Intl. Workshop on Sensor Nets. & Apps, Sept 2002, pp. 112–121 (2002)
33. Costa, J.A., Patwari, N., Hero III, A.O.: Distributed multidimensional scaling with adaptive weighting for node localization in sensor networks. IEEE/ACM Trans. Sens. Net. **2**(1), 39–64 (2006)
34. Albowicz, J., Chen, A., Zhang, L.: Recursive position estimation in sensor networks. In: IEEE Int. Conf. Net. Proto. Nov 2001, pp. 35–41 (2001)
35. Savvides, A., Han, C.C., Srivastava, M.B.: Dynamic fine-grained localization in ad-hoc sensor networks. In: Savvides, A., Han, C.C., Srivastava, M.B. (eds.) Proceedings of the Fifth International Conference on Mobile Computing and Networking (Mobicom 2001), Rome, Italy, July 2001, pp. 166–179
36. CC2431: System-on-Chip for 2.4 GHz ZigBee/IEEE 802.15.4 with Location Engine. http://www.ti.com/lit/gpn/cc2431 (2005)
37. Ihler, A.T., Fisher III, J.W., Moses, R.L.: Nonparametric belief propagation for self-calibration in sensor networks. In: IEEE ICASSP 2004, vol. 3, pp. 861–864 May (2004)
38. Lazos, L., Poovendran, R.: SeRLoc: secure range-independent localization for wireless sensor networks. In: WiSe '04: Proc. 2004 ACM workshop on Wireless security, pp. 21–30 (2004)

Location Anonymization

► Cloaking Algorithms for Location Privacy

Location-Aware

► Time Geography

Location-Aware Access Control

► Security Models, Geospatial

Location-Aware Technologies

Harvey J. Miller
Department of Geography, University of Utah,
Salt Lake City, UT, USA

Synonyms

Position-aware technologies; Global positioning system; GPS; Location based systems LBS

Definition

Location-aware technologies are devices that can report their precise geographic location at high temporal resolutions.

Main Text

Location-aware technologies (LATs) are devices that can report their geographic location in near-real time using methods such as *radiolocation* that exploit wireless communication systems, the *global positioning system* (GPS) that utilize time differences of signals from satellites in Earth orbit and *dead-reckoning* that use distances and directions along a route from a known location to determine the current location.

An emerging LAT is *radiofrequency identification* (RFID) tags. Mobile RFID tags transmit data to fixed readers using either passive (powerless) or active (powered) methods. Active tags are heavier and more expensive, but have a longer range and cheaper readers than can track multiple tags simultaneously. Unlike GPS, RFID tags must self-identify to the reader since the reader (not the client) conducts the location calculations. This means that RFID systems have a greater potential for surveillance

LAT enable *location-based services* (LBS). LBS provide targeted information to individuals based on their geographic location though wireless communication networks and devices such as portable computers, personal digital assistants (PDAs), mobile phones and in-vehicle navigation systems.

Cross References

► Geographic Knowledge Discovery
► Time Geography

Recommended Reading

1. Benson, J.: LBS technology delivers information where and when its needed, Bus. Geogr. **9**(2), 20–22 (2001)
2. Grejner-Brzezinska, D.: Positioning and tracking approaches and technologies. In: Karimi, H.A., Hammad, A. (eds.) Telegeoinformatics: Location-based Computing and Services, pp. 69–110. CRC Press, Boca-Raton, FL (2004)
3. Morville, P.: Ambient Findability: What We Find Changes Who We Become. O'Reilly Media, Sebastopol, CA (2005)

Location Based Services

► Indoor Positioning with WirelessLocal Area Networks (WLAN)
► Nearest Neighbor Queries in Network Databases
► Privacy Threats in Location-Based Services

Location-Based Services

► Cloaking Algorithms for Location Privacy
► Indoor Positioning
► Information Services, Geography
► Moving Object Uncertainty

L

Location-Based Services: Practices and Products

Joseph R. Francica
Directions Media, Huntsville, AL, USA

Synonyms

Services, web; Service, location; Positioning

Definition

Location Services or Location-based services (LBS) are terms that have been used to describe wireless, mobile, and mostly handheld devices that use wireless communications to deliver information that conveys a geographical reference. As such, information delivered to users of cellular phones, for example, depends on the use of a device that is capable of utilizing location-determination technology such as Global Positioning Systems or Wi-Fi to calculate a specific latitude and longitude of the device. This is a fundamental premise of LBS technology. Location becomes a supporting attribute to ancillary information related to that location such as a street address or other point of interest. Typical applications include automatic location identification (ALI) for emergency response phone calls made to a 9-1-1 dispatch operator.

However, an expanded use of the term is offered to include enterprise computing solutions of which location determination is an essential part. For example, corporations that utilize wireless, location-aware devices to track its fleet of trucks or cars and to mobilize this fleet more efficiently are using a location-based service. Likewise, cellular phone users who subscribe to an "in-network" plan that allows them to determine the location of friends and family are utilizing a form of location-based social networking.

Terms & Types

LOCATION SERVICE: A location service, in the broadest sense, is any service or application that extends spatial information processing ... to end users via the Internet and/or wireless network [1].

Adapted from Niedzwiadek [2], here are definitions (List 1) of information types and a list of examples of location services (Table 1), in the form of user requirement statements (or questions), organized by information type (row) and market segment (column).

List 1: Types of Information in Time or Space [2]:

Positions. Fixed locations. Expressed in terms of coordinates, positions on a map, named places, and so forth.

Events. Time-dependent incidents (past, present, or future) at one or more locations.

Distributions (e. g. demographics). The densities, frequencies, patterns, and trends of people, objects or events within a given area(s).

Assets. Fixed and/or mobile assets. Asset management. Inventories. Condition/status.

Service Points. Points of service delivery. May also pertain to prospects or targets of interest. Further characterized by levels of service and quality of service.

Routes. Navigational information expressed in terms of coordinates, directions (angles), named streets and distances, landmarks, and/or other navigation aids. Navigational logs.

Context/ Overview. Maps, charts, three-dimensional scenes (virtual reality) or other means for representing the context and relationships between people, objects and events over a given area(s).

Directories. Catalogs. Listings. Directories.

Transactions. Transactions for the exchange of goods, services, securities, etc. Trading services. Financial services.

Sites. Characteristics of a given site (e. g. suitability).

Historical Background

One of the key drivers of location-based service was the 1996 mandate of the Federal Communications Commission of the United States to have all cellular carriers identify the location of a 9-1-1 caller. This mandate is being fulfilled in stages but over a longer period of time than first anticipated as the cellular companies did not have the infrastructure or business plan to facilitate the service. The Phase I plan required cellular carriers to locate callers within the vicinity of the cellular antenna. Phase II requires that operators locate cellular calls to 9-1-1 to within 50 to 300 meters. (See FCC http://www.fcc.gov/911/enhanced/). Users of location services have been hindered by the lack of devices that are capable of recognizing the position of a user. Early location-based services depended on the user to input their location to a wireless device. The user would typically manually identify their location by typing a street address or intersection to their wireless device and the service would search for certain services within that vicinity. It is expected that as more wireless devices are able to automatically determine the user's position by incorporating GPS chips or using 802.11 protocols (W-Fi) by triangulation that the value proposition to users will become more evident.

Scientific Fundamentals

The basic technology infrastructure to support location-based services is supported by:

1. Location positioning infrastructure – This would include the navigational satellite constellations of the United States Global Positioning System (GPS), the European Galileo system, and the Russian Global Navigation Satellite System (GLONASS). Other positioning infrastructure is now being tested and deploying using the 802.11 Wi-Fi protocols.
2. Location determination – Mobile devices need to include technology that utilizes the location positioning infrastructure. These devices will most often include a receiver that will seek signals from three or more global positioning satellites to determine a location. The receivers interpret the time difference of arrival (TDOA) of the signal from each satellite and calculate the device position, often referred to as multilateration. Certain technology providers such as Skyhook Wireless utilize a database of the location of known Wi-Fi access points to establish position. Skyhook and GPS chip manufacturer SiRF Technology are partnering to provide both GPS and Wi-Fi positioning on a single chip. In areas where it is difficult to obtain a satellite fix, such as in an urban canyon or indoors where GPS will not function, Wi-Fi can be used as an alternative location determination technology.
3. Application Servers – Once the geographic position of the receiver is established, the location of the device is passed to a computer or geographic server that utilizes the location information for application development.

Location-Based Services: Practices and Products, Table 1 Types of Location Services [2]

Types of Location Information	Location Services		
	Consumer	**Business**	**Government**
Positions	Where am I? (map, address, place) Where is? (Person, business, place . . .)	Contact nearest field service personnel. Where is this business located?	Location-sensitive reporting. What's your 20?
Events	Car broken down . . . need help. Medical alert!	Local training announcements. Traffic alert!	Local public announcements. Accident alert!
Distributions	House hunting in low density area. Vacationing near highest concentration of . . .	High growth trend? Sales patterns?	Growth patterns? Per capita greenspace?
Assets	Where is my car? Lowest insurance rates?	Where are my dispatched repair trucks? Status of my holdings?	Where are the snowplows? Road maintenance
Service Points	Tell me when I'm near where I'm going. Where are the sales?	Where are my customers, given target profile? Targeted advertising	Economic development areas? New zoning
Routes	How do I get there? (address, place) Fastest route (given traffic situation)?	Best delivery route given shipping manifest, traffic and weather? Taxi dispatch	Traffic patterns? Emergency dispatch
Context (Overview)	Nearest visible landmark? Show me the nearest___ (business, place . . .)	What's near the hotel? Show me car rentals near the airport	Collaborative economic planning. Local commerce
Directories	Looking for nearest___ (specialist . . .) Where can I buy? (product, service)	Best supplier within next two hours? Nearest repair services?	Public services. Outsourcing?
Transactions	Lowest shipping rates? Must purchase in specific location	Low cost distribution services? Location-sensitive quick-dial	Tax revenues. Location-sensitive tolls
Sites	Candidate properties to build my house. Places to visit?	Candidate store sites? Optimum cell tower locations?	New schools? Environmental monitoring stations?

Geographic information systems (GIS) or specialized location servers are tasked with geocoding the information to reveal additional information such as an address, for example.

Key Applications

Emergency 911/Automatic Location Identification

Calls to a police or fire agency, often referred to as a public service answering point or PSAP, via the Emergency 911 number must respond to an incident at a specific location. Emergency service dispatchers have had the ability to "look-up" the street address of a fixed, wireline service for many years. This is often called Automatic Number Identification/Automatic location identification (ANI/ALI). This service as applied to wireless communication devices relies on location determination technology. This technology may be found at either the cellular tower base station or on the client side device such as a cell phone. At the tower base station, technology is employed to support location determination by triangulating the position of a call based on the location of three or more cell towers and the time delay of arrival (TDOA) of the signal from the mobile handset. On the client device, GPS is often used to accurately determine the position of the device which is then broadcast to the dispatcher.

Logistics and Transportation: This application applies to a variety of services that require vehicles and cargo to be managed efficiently. Often this application requires both a wireless device for drivers and a server-based or web-based solution for call center operators and dispatchers to monitor the location of personnel and products. The two applications below describe examples of LBS:

Field Service Management (FSM): Government agencies and corporations that deploy mobile field personnel who provide maintenance, repair or delivery services will utilize wireless devices in conjunction with a desktop software or web service for dispatching, routing, and tracking. Forecasting service demand is a key element of FSM. It is necessary for cutting excess idle time in the field workforce. Another element of FSM helps to determine the best deployment strategy to maximize the coverage of personnel workload in a service area, and to minimize the cost to do so. As a result of understanding these elements, managers will better understand and evaluate the actual effectiveness of the optimization routines and re-evaluate further forecasting improvements.

Mobile Asset Management: Certain parcels, cargo, or vehicles have a need to be tracked. The locations of these assets that are in transit provide managers the ability to alter routes and monitor deliver progress. These items will utilize location-determination technology such as GPS receivers or RFID tags to identify position.

Personal Navigation and Concierge Services: Many GPS manufacturers, cellular telecommunications companies, and web-service providers are offering portable devices with GPS to request driving directions and local search information on nearby services (e. g. gas stations, movie theaters, restaurants, car dealerships, repair services, etc.). One of the more popular services that helped spawn this industry was OnStar (offered by General Motors), which allowed the driver of a vehicle to push a button to have direct communication with a call center that offered immediate assistance with driving directions as well as emergency roadside assistance. In addition, should the vehicle be engaged in an accident where the driver is unable to communicate with the OnStar call center, OnStar is connected to some of the vehicle's electronics and will be notified if air bags are deployed. The OnStar call center will then notify emergency responders via a 911 call to dispatch roadside assistance.

Telematics: This term is a blend of two words: telecommunication and informatics: the linking of mobile things (vehicles, people, cell phones, PDAs, etc.) to remote sources of information and entertainment [3].

Social Networking: This service allows cellular phone subscribers the ability to locate friends or family. Some services will alert the subscriber when a member of their network enters a certain geographic proximity.

Location-based Gaming: Using a wireless device, subscribers will be issued instructions to locate caches of prizes or other instructions to pursue further information.

Inventory Tracking & RFID: Tracking inventory carried by truck often only allows product manufacturers to know only basic information about the gross cargo load. Radio Frequency Identification (RFID) tags applied to palettes or single items is a technology that allows for more comprehensive information to be stored on the tag about the product in which it is attached. RFID readers, generally a stationary device, will emit a radio signal that will "illuminate" the RFID tag and scan information contained on the tag for information about the item or palette as it passes the reader. Readers may be positioned at various locations. Those locations are known to the manufacturer and a time stamp can be applied to the information at the RFID-tagged item passes each reader thus allowing logistic managers to keep a more accurate inventory.

Fitness and Exercise: Increasingly, more devices and Mobile Virtual Network Operators (MVNO) are providing services specifically for the exercise and fitness market. Devices released by Trimble Outdoors and services for monitoring biking, hiking and running activities offered by Boost Mobile, an MVNO and subsidiary of Sprint Nextel, are gaining popularity in a niche market.

Future Directions

Berg Insights, a noted market research firm, has the following analysis of the LBS market:

> "Penetration for fleet management technology in the European Union road transportation industry will exceed 70 percent by 2010. The report urges the industry to look beyond transportation for additional growth opportunities. Until very recently most vendors have focused exclusively on road transport enterprises, even though these only represent 11 percent of the total potential market. There are twice as many commercial vehicles in the construction industry, but no one seems yet to have figured out how to reach that segment. Because most other segments will remain relatively undeveloped the total market penetration for fleet management technology in Western Europe will be significantly lower at about 15 percent in 2010."

Berg Insight also forecasts that the major truck manufacturers – starting with Volvo and Mercedes-Benz – will make satellite tracking and mobile data communication standard issue on high-end models from as early as 2008. Berg Insight pointed out that remote monitoring is already a standard feature for a certain Volvo truck engine model in the US and said he believes this is the start of a very significant development in the commercial vehicle industry [3].

Another market research firm estimates the global potential revenue of the LBS market to be $750 Billion by 2019 [5].

Cross References

- ▶ Indoor Positioning
- ▶ Positional Accuracy Improvement (PAI)
- ▶ Radio Frequency Identification (RFID)
- ▶ Road Maps, Digital
- ▶ Routing Vehicles, Algorithms

Recommended Reading

1. Koppell, I.: What are Location Services? – From a GIS Perspective, LBS360.NET (http://www.lbs360.net/LBSArticles/ESRI.What%20are%20LS%20Whitepaper.pdf)
2. Niedzwiadek, H.: Where is the Value in Location Services, LBS360.NET. (http://www.lbs360.net/LBSArticles/HN.Where%27s%20the%20Value%20in%20Location%20Services.pdf), July 1 (2002)
3. Berg Insight, Press Release, Directions Magazine, November 15, (2006)
4. Francica, J.: LocationBI – Location-based Business Intelligence, Directions Magazine, October 22 (2003)
5. Niedzwiadek, H.: Where's the Value in Location Services? LBS360.NET, July 1 (2002)
6. Reid, H.: Telematics Detroit, Directions Magazine, May 22 (2005)
7. Williams, D.: The Definitive Guide To GPS, RFID, Wi-Fi, and Other Wireless Location-Based Services – What Applications Will Succeed in What Markets, and How To Design and Market Them (2005)

Location Based Systems

▶ Location-Aware Technologies

Location Blurring

▶ Cloaking Algorithms for Location Privacy

Location Estimation

▶ Indoor Positioning, Bayesian Methods

Location Intelligence

DEVON WOLFE[1], GEORGE MOON[2]
[1] MapInfo Canada, Toronto, ONT, Canada
[2] MapInfo Corporation, Troy, NY, USA

Synonyms

Locational data analysis

Definition

Location intelligence is defined as the use of locationally-referenced information as a key input in business decision making. Location intelligence uses geographic information systems (GIS) tools and techniques to transform and analyze data that becomes valuable information to make more informed and rational business decisions. Thus, location intelligence is neither simply an analytical technique nor business process; but a discipline that encompasses both.

Historical Background

Location intelligence is a term that has only recently come into wide usage in the business world. The term is derived from the concept of business intelligence, which seeks to leverage a corporation's data to make business decisions. Location intelligence expands this concept by adding a spatial perspective to business data analysis and decision making.

Examples of locationally-referenced business data include such items as property locations, customer locations, or supplier locations. These data may be found in the form of addresses, geographic coordinates, region designations (postal code, political boundary).

These data can be combined with other types of geographic data such as population, road networks, climate information, or topography to analyze various spatially-referenced phenomena.

Scientific Fundamentals

To develop a location-intelligent solution, there are generally four recognized processes: identification of locational elements, transformation of those elements into a form suitable for analysis, development of research plans for how these data can be queried and analyzed, and use of statistics and other methods to draw conclusions from the data.

Identification of Locational Elements

First, databases (or data sources) must be reviewed for data that could contain a locational element. As locational elements are identified, the database analyst must determine what type of locational element the data represents. Most locational data used for analysis can be classified as either point, linear, or polygon types. Polygons are often referenced for the business community as regions.

In many cases locational elements are added to a database or referenced from the database for use in location intelligence. Often valuable insights result from using complementary data that extend what an organization typically captures. In the case of retail for example, the geodemographic or 'lifestyle' characteristics of a customer significantly enhances the understanding of retail performance. http://www.geoplace.com/uploads/FeatureArticle/0411ei.asp

Transformation of Location Elements

Database or datawarehousing personnel use techniques called extract, transform and load (ETL). The transformation of location elements is a spatial ETL. This process varies according to the type of locational data classification as noted above. In the case of point data, this may mean converting an address into a point data type using geocoding software; a regional designation such as sales territory would be matched to a corresponding polygon file, to create a summary by region; for data such as a highway represented as a linear data type buffers to look for nearby spatial relationships, impedances on speed, and the like can be used to enhance the value of the data.

Development of Research Plans

After data have been identified and transformed into forms suitable for display and analysis, potential users conduct a needs analysis to determine how these data may be used to answer key business questions. From this analysis, a research plan can be developed that will guide the business in how to display, analyze, and use locationally-referenced data. This plan should outline data sources to be used and linkages between them, the business problem to be solved, specific techniques necessary to analyze the data, and the outcome or deliverables that will be provided to endusers. Often, experienced database analysts, business analysts, researchers, and statisticians are required to complete this step.

One of the most important aspects of business data is its dynamic nature. Location intelligence, through exploitation of data warehouses and constant data updating, can take advantage of dynamic business data to show movement in not just time, but in space as well. The combination of the spatial and temporal elements is extremely powerful for business decision making. However, as in most data-driven research, the key to a quality process is the quality of the input data. To be effective in real time, quality must pervade the process of acquiring, cleansing, updating, and verifying data. Thus, extensive planning for data quality control is an essential part of the research planning process.

Methods to Draw Conclusions from Location Data

There are a variety of ways to use location data in business analysis. The simplest is to map relevant data to create a visual representation of the interaction of various geographic phenomena in a way that transcends the limitations of data in tabular form. In many cases, this simple step allows users to make inferences that would never have been imagined if the data were not depicted in visual form. Next, analytical operations can be used to interact with the data on the map. These operations may include spatial queries that depict certain subsets of the data to reveal spatial patterns; thematic mapping to explain how specific attributes vary over space; and spatial boundary creation, such as radii, drive times, buffers, and other geographic boundaries that allow data to be summarized and analyzed according to locational proximity to a point, line, or region. These analytical operations help to reveal spatial relationships in both a visual and a summary form. For example, determining and thematically-displaying the population density can be useful to understanding the locations of concentrations of people. Selecting geographic objects can also be done and then used as a filter back into the database or data warehouse.

These techniques are commonly known as descriptive analytics, because they allow a user to analyze phenomena using transformations that describe certain variables. These methods, however, do not fully exploit the richness of the underlying relationships and interactions between various locationally-referenced attributes.

A more rigorous and scientific form of analysis is to use the concept of predictive analytics to determine key relationships between variables using statistical or data mining processes, and then to develop predictive models that forecast outcomes given changes in locationally-referenced variables.

The addition of spatial extensions to database management systems (DBMS) means that it is possible to use the security and scalability of a DBMS to perform spatial operations on data inside the database before returning the results. Combining spatial and non-spatial capabilities is a powerful way return only data that is truly relevant for presentation or further analysis.

Key Applications

Retail

Retail businesses serve as distribution points for goods to customers. As such, retailers must carefully plan locations to be the most advantageous and efficient forms of distribution for their key audience. In application, this means that retailers can use location intelligence to:

- Determine a specific forecast projection for a new store
- Determine optimal store locations
- Simultaneously maximize market share and per store performance
- Quantify and minimize sales transfer between stores

- Precisely match media and marketing messages to targeted households
- Determine optimal delivery routes from warehouses to stores, or from stores to consumers
- Understand, diagnose and recommend solutions for underperforming stores

Insurance

Insurers must manage their operating territories to provide an efficient distribution of sales agents and claims representatives. At the same time, insurers are also concerned with the risk inherent in the physical location of an insured property, whether it is a business, a home, or an automobile. Insurers use location intelligence to:

- Accurately assess marketing potential, better focusing, marketing, sales, and distribution management, and increasing producer effectiveness
- Improve underwriting decisions by providing more accurate exposure analysis
- Increase competitiveness through more refined and accurate pricing
- Improve customer satisfaction by eliminating unnecessary questions that can be answered with location intelligence
- Increase organizational efficiency through the use of real-time rules processing that returns a casualty rating from spatial queries to databases containing property locations and hazard occurrences
- Manage risk on a portfolio basis and comply with regulatory reporting requirements
- Determine where to deploy claims adjusters in the case of a natural disaster

Financial Services

Banks, credit issuers, and other financial services firms seek to maximize the profitability of financial transactions and loans, while at the same time holding down distribution and sales costs. These institutions use location intelligence to:

- Maximize branch performance
- Evaluate expansion opportunities by determining the optimal number, placement, and priority of new locations
- Optimally allocate sales and branch staff
- Understand customer needs and behaviors
- Determine product mix according to the geographic location of the customer
- Identify underperforming markets, areas, or branches and determine strategies to improve or, if necessary, exit markets

Communications

Wireless companies need to match coverage and signal strength with the location of customers who continually change location. In addition, they need to understand consumer purchasing behavior to develop strategies to reach the right customer in the right place. Communications companies use location intelligence to:

- Analyze market demand, network coverage, and competitor data to optimize network design, build-out, and maintenance
- Provide superior customer service, including identification of emerging trouble spots, calculation of downtimes, and real-time deployment of network engineers
- Understand customer demand and competitive threats to develop market-driven product offerings and competitive pricing schemes
- Locate customer service outlets and recruit third-party distributors to increase market share and customer service
- Generate highly-qualified sales leads based on service availability and a customer's likelihood of subscribing

Government

Governmental bodies must plan, fund, and provide services to many different constituencies. Location intelligence can help these agencies to:

- Attract, retain, and support local businesses in order to create jobs and strengthen the tax base
- Plan and develop large-scale public works projects
- Evaluate the need for and effectiveness of national government aid resources in human services, economic development, agriculture, and other fields
- Enhance disaster forecasting and emergency preparedness and recovery operations
- Use location information of crime events with predictive capabilities to deploy more effective police patrols
- Improve predictive capabilities for national and homeland security functions

Future Directions

Location intelligence is still young. It has gone beyond the 'early adaptor' stage for businesses however there will be significant adoption to come. As more and more organizations use location intelligence in their everyday processes, the emergence of innovative new ideas that leverage the spatial element of business data is expected. Ultimately, location intelligence will be an integral part of business decision making in much the same way as other data-oriented business processes are now.

Recommended Reading

1. Location Intelligence: The New Geography of Business, Business Week Research Services (2006)
2. Chen, A.: MasterCard Cashes in on Web Services. http://www.eweek.com/article2/0,1759,1761448,00.asp (2005)
3. Analytical CRM software opens up new worlds to marketers, by Paul Thompson, June 17 (2002) http://www.strategymag.com/articles/d+i/20020617/dicrmanalytical.html?word=Analytical&word=CRM&word=Software
4. Diversity in America, Supplement to American Demographics, Alison Stein Wellner, November, (2002) http://resource.mapinfo.com/static/files/document/1074653455507/diversity.pdf and www.demographics.com
5. MAPS AND MANAGEMENT: COMSTAT EVOLVES, Raymond Dussault, Editor, Government Technlogy, April (2000) http://www.govtech.net/magazine/crimetech/Apr00/CTEComstat.php
6. Site Selection: New Advancements in Methods and Technology, Buckner, Robert W., National Book Network (1998)
7. A status report on location analysis – retail location analysis – Guest Column – Column, DSN Retailing Today, Jan 26, 2004 by David Rogers http://www.findarticles.com/p/articles/mi_m0FNP/is_2_43/ai_112792451

Location Perturbation

► Cloaking Algorithms for Location Privacy

Location-Privacy

► Cloaking Algorithms for Location Privacy

Location Server

► Privacy Threats in Location-Based Services

Location Tracking

► Indoor Positioning, Bayesian Methods

Location Trusted Server

► Privacy Threats in Location-Based Services

Locational Data Analysis

► Location Intelligence

Locking

► Smallworld Software Suite

LOD

► Hierarchies and Level of Detail

Logic Programming Language

► Constraint Database Queries

Long-Running Spatio-temporal Queries

► Continuous Queries in Spatio-temporal Databases

Lossless Image Compression

► Image Compression

Lossy Image Compression

► Image Compression

Machine Readable Geographic Data

► Feature Catalogue

Magik, Smallworld

► Smallworld Software Suite

Management of Linear Programming Queries

► MLPQ Spatial Constraint Database System

Manhattan Distance

► Distance Metrics

Manifold

► Spatial Data Transfer Standard (SDTS)

Manifold Rules

► Smallworld Software Suite

Manifolds

► Smallworld Software Suite

Mantel Test

► CrimeStat: A Spatial Statistical Program for the Analysis of Crime Incidents

Map Accuracy

► Spatial Data Transfer Standard (SDTS)

Map, Bi-Plot

► Geographic Dynamics, Visualization And Modeling

Map, Centrographic Timeseries

► Geographic Dynamics, Visualization And Modeling

Map Data

► Photogrammetric Products

Map Distribution

► Data Infrastructure, Spatial

Map Generalization

ANNE RUAS
Laboratory COGIT, IGN-France, Saint-Mandé, France

Synonyms

Generalization

Definition

Map generalization is the name of the process that simplifies the representation of geographical data to produce a map at a certain scale with a defined and readable legend. To be readable at a smaller scale, some objects are removed; others are enlarged, aggregated and displaced one to another, and all objects are simplified. During the process, the information is globally simplified but stays readable and understandable.
The smaller the scale, the less information is given per square kilometer. Conversely, the larger the scale, the more detailed is the area mapped for the same map size. For

a given size of map sheet, nearly the same quantity of information is given for different scales, either privileging the density of field information (for larger scale) or the spatial extension (for smaller scale).
This process is used both in manual and digital cartography.

Main Text

Generalisation can be first defined by means of graphical constraints and scale. On a map the information is represented by means of symbols. These symbols are coded representations which ensure the interpretation of the meaning. These symbols have minimum sizes that ensure not only good perception but also the recognition of the symbols and their associated meaning. As an example, a very small black polygon representing a building will be seen as a dot and not a building. In the same way two symbols too close one to another will be seen as a single symbol even if in the real world they represent two different entities. These graphical constraints are called legibility or readability constraints. When represented on a map the graphical objects are not a faithful representation of the entities' sizes at a given scale but are symbolic representations which maximize communication of information.
In order to respect graphical constraints, some objects are enlarged and some are displaced one to another. These geometric distortions are minor at the large scale (1:2,000–1:10,000), common at the medium scale (1:15,000–1:50,000) and frequent and very large at small scales. As an example a 6 m width road represented by a line of 0.6 mm on a map is enlarged 10 times at 1:100,000 and 100 times at 1:1,000,000! Of course, when the scale decreases it is physically not possible to enlarge and displace all objects: many objects are removed, some are aggregated, and they are all simplified. These operations of enlargement, displacement, selection, aggregation and geometric simplifications are the main operations of generalization. There is a good large amount of literature devoted to defining and specifying the operators and algorithms of generalization.
But generalization can also be defined through a more geographical view point. A geographical representation—being a map or a data base—represents the real world at a certain level of detail. This level of detail implicitly defines the type of information represented and their reasonable use. Geographical phenomena have a certain size and they can only be depicted within a certain scale range. Generalization is a process that aggregates and simplifies the information in order to make apparent more general concepts through their representation. When scale decreases some concepts disappear while others appear. The transition between scales can by smooth and continuous for some themes or abrupt for others. Generally networks are preserved but their density decreases, while small size objects such as houses are changed into urban areas, dots and even nothing. Müller [2] speaks of change under scale progression (CUPS) to characterize these transformations through scale.
The complexity of the automation of the generalization process is due to the diversity of geographical information and its contextual nature. Generalization should preserve relationships and properties that are, most of the time, implicit. Current models of generalization rely on optimization techniques or on a multiagent system paradigm.

Cross References

► Abstraction of GeoDatabases
► Generalization and Symbolization

Recommended Reading

1. McMaster, R., Buttenfield, B.: Map Generalization. Longman, Harlow (1991)
2. Müller, J-C., Lagrange, J.-P., Weibel, R.: GIS and Generalization–Methodology and Practice. Taylor and Francis, London, UK (1995)
3. Mackaness, W., Ruas, A., Sarjakoski, L.T.: Generalisation of Geographic Information: Cartographic Modelling and applications. Elsevier, Burlington, MA, USA (2007)

Map Overhaul

► Positional Accuracy Improvement (PAI)

Map Quality

► Positional Accuracy Improvement (PAI)

Map-Matching

DIETER PFOSER
RA Computer Technology Institute, Athens, Greece

Definition

Sampling vehicular movement using GPS is affected by error sources. Given the resulting inaccuracy, the vehicle tracking data can only be related to the underlying road network by using *map-matching* algorithms.

Main Text

Tracking data is obtained by sampling movement, typically using GPS. Unfortunately, this data is not precise due to the *measurement error* caused by the limited GPS accuracy, and the *sampling error* caused by the sampling rate, i. e., not knowing where the moving object was in between position samples. A processing step is needed that matches tracking data to the road network. This technique is commonly referred to as *map matching*.

Most map-matching algorithms are tailored towards mapping *current positions* onto a vector representation of a road network. Onboard systems for vehicle navigation utilize dead reckoning besides continuous positioning to minimize the positioning error and to produce accurate vehicle positions that can be easily matched to a road map. For the purpose of processing tracking data, the *entire trajectory*, given as a sequence of historic position samples, needs to be mapped. The fundamental difference in these two approaches is the error associated with the data. Whereas the data in the former case is mostly affected by the measurement error, the latter case is mostly concerned with the sampling error.

Cross References

► Dynamic Travel Time Maps
► Floating Car Data

Recommended Reading

1. Brakatsoulas, S., Pfoser, D., Sallas, R., Wenk, C.: On map-matching vehicle tracking data. In: Proc. 31st VLDB conference, pp. 853–864 (2005)

Mapping

► Public Health and Spatial Modeling

Mapping and Analysis for Public Safety

► Hotspot Detection, Prioritization, and Security

Maps, Animated

► Geographic Dynamics, Visualization And Modeling

Maps On Internet

► Web Mapping and Web Cartography

MapServ

► University of Minnesota (UMN) Map Server

MapServer

► Quantum GIS
► Web Feature Service (WFS) and Web Map Service (WMS)

MapWindow GIS

DANIEL P. AMES, CHRISTOPHER MICHAELIS,
ALLEN ANSELMO, LAILIN CHEN, HAROLD DUNSFORD
Department of Geosciences, Idaho State University,
Pocatello, ID, USA

Synonyms

Open-source GIS; Free GIS; Programmable GIS components; .NET framework; ActiveX components

M

Definition

MapWindow GIS is an open source geographic information system that includes a desktop application and a set of programmable mapping and geoanalytical components. Because it is distributed as open source software under the Mozilla Public License, MapWindow GIS can be reprogrammed to perform different or more specialized tasks and can be extended as needed by end users and developers. MapWindow GIS has been adopted by the United States Environmental Protection Agency as a platform for its BASINS watershed analysis system and is downloaded over 3000 times per month by end users who need a free GIS data viewer and programmers who need tools for commercial and non-commercial software applications.

Main Text

MapWindow GIS is a desktop open source GIS and set of programmable objects intended to be used in the Microsoft Windows operating system. Because it is developed using the Microsoft .NET Framework, it is optimized for the Windows environment and can be extended by programmers using the Visual Basic and C# languages. The MapWindow GIS desktop application plug-in interface supports custom tool development by end users and the MapWindow Open Source team. Additionally, software developers can use the core MapWindow ActiveX and .NET programming components to add GIS mapping and geoprocessing functionality to custom standalone applications.

MapWindow GIS has been adopted by the United States Environmental Protection Agency, United Nations University and others as a development and distribution platform for several environmental models including BASINS/HSPF, FRAMES-3MRA and SWAT. Other users and developers have modified and applied MapWindow GIS for use in the fields of transportation, agriculture, community planning and recreation. MapWindow GIS is continually maintained by an active group of nearly 50 student and volunteer developers from around the world who regularly release updates and bug fixes through the www.MapWindow.org web site.

Cross References

► Open-Source GIS Libraries

Marginalia

► Metadata and Interoperability, Geospatial

Market and Infrastructure for Spatial Data

PETER KEENAN
UCD Business School, University College Dublin, Dublin, Ireland

Synonyms

Spatial information; Geographic information

Definition

GIS combines hardware and software to allow the storage and manipulation of spatial data. GIS applications typically combine specific spatial data with data on the general geographic features of an area. The availability and cost of this general data is an important factor in the more widespread adoption of spatial techniques. The spatial data market is the system enabling the transaction of business between buyers and sellers of such data. This system must reward those who collect spatial data and protect their rights over that data while making the data available to users under reasonable conditions.

Historical Background

GIS applications are comparatively resource-intensive, as they require more complex software and consequently more powerful hardware than many common applications of Information Technology (IT). However, the rapid development of IT means that sophisticated GIS software is now readily available and that the computer technology required for this software is now a small proportion of cost of using GIS. Nevertheless, while hardware and software developments have been favorable to the widespread use of GIS, the ready availability of spatial data is also a critical requirement for greater GIS use. In traditional IT applications for example, business data processing, most of the data used originates within an organization and is often the result of the use of the technology. GIS was typically first employed by organizations with existing non-computerized spatial data and was used to build and manage large computerized spatial data-sets. However, as GIS use has extended to a wider user community, many potential spatial applications have a much smaller proportion of unique data collected for that specific application and organization. This internal data, collected for a particular project, only becomes useful when combined with existing general geographic data sourced outside the organization. For example, information on the spatial location of retail outlets may only become useful when linked with other spatial data on shared transportation networks and demographic data for the region. Many organizations, across a wide variety of sectors, will operate in the same geographic space. These will require similar data on the general geography of the region in which they operate; this can be combined with their own spatially referenced datasets to meet their specific needs. Such organizations may have little in common, other than their shared location in the same geographic region and therefore a need for spatial data about their area of operations. Consequently, applications as diverse as fuel delivery, mobile phone service provision and emergency service dispatch will have in interest in using the same spatial data.

Scientific Fundamentals

The first requirement for spatial data availability is that it must be available in an appropriate format for the varied range of applications that might arise in a geographic region. While a large amount of spatial data has already been collected for developed countries, this may not be in a format suitable for the full range of possible applications for that data. In most cases traditional government mapping agencies, such as the Ordnance Survey in the UK or the Bundesamt für Kartographie und Geodäsie in Germany, now have complete digital databases of their cartographic resources. However, the traditional focus of these agencies has been on the production of maps, rather than the use of spatial data for computer processing, and the organization of their spatial database reflects this orientation. New technologies such as remote sensing and global positioning systems (GPS) have allowed other private sector organizations collect large amounts of spatial data. Private sector spatial data providers are likely to more aware of the need to organize their data in a way which will make it useful to the maximum number of potential users. Local, regional and national public authorities will generate spatial data for the areas under their administration. However, these public organizations will collect and represent their spatial data in a way suitable for their own needs, which may not be ideal for other potential users. Therefore, much of the spatial data needed for GIS applications has already been collected, but it is not always in a format suitable for all potential users. However, the existence of spatial data is only the starting point for its widespread use. There are at least two further requirements: potential users of the data must be aware that it is available and the data must be provided at an economic cost.

While there are challenging technical issues relating to synthesis of spatial data, there are a number of initiatives to address these problems. Several countries have launched spatial data infrastructure (SDI) initiatives to provide a framework for better integrating the collection and use spatial data [1]. SDI frameworks may include spatial data clearinghouses, which can be defined as a computerized facility for searching, viewing, transferring, ordering, advertising, and disseminating spatial data across the Internet. In the US, the Federal Geographic Data Committee defines the United States national SDI as an umbrella of policies, standards, and procedures under which organizations and technologies interact to foster more efficient use, management; and production of geospatial data [2]. In the US, the SDI has evolved to better integrate all forms of public data. The objective is to make this data available over the Internet and the Geospatial One-Stop (GOS) initiative (www.geo-one-stop.gov/) aims to provide a single point of access to this data. Consequently, SDI initiatives can be integrated with a geoportal which provides distributed access to the spatial data. Geoportals can primarily provide access to spatial data or can provide services which provide the result of spatial processing [3]. While it may take some time, these initiatives suggest that appropriate technical standards will emerge and that these challenges can be successfully overcome [4].

The indexation of suitable data sources can exploit the fact that spatial data has a unique location reference and can be organized with respect to this location. Consequently, an index can be produced of all data, spatial and non-spatial, relating to a particular spatial location. A geolibrary is one approach to the organization of information in a spatially referenced structure. A geolibrary allows search for content, both spatial and non-spatial, by geographic location in addition to traditional searching methods. The concept of a geolibrary originated in the 1990s [5] and the Alexandria digital library at the University of California, Santa Barbara is generally regarded as the first major prototype geolibrary (webclient.alexandria.ucsb.edu/). The geolibrary model was further defined within the GIS community by the Workshop on Distributed geolibraries: Spatial Information Resources, convened by the Mapping Science Committee of the US National Research Council in June 1998. Geolibraries are based on the spatial referencing of data; this data need not be either digital or spatial in nature. However, the geolibrary concept is especially useful as a means of organizing spatial data [6].

In principle, geolibraries should provide access to all types of data, both spatial and non-spatial. These resources should be centrally indexed, even if the physical storage is distributed. As more spatial data becomes available, this allows a vast number of potential combinations of these data-sets, facilitating a wide range of applications. However, there must be a clear economic incentive for spatial data owners to make their data available to other potential users. Spatial data may originate from organizations that see themselves as spatial data suppliers. These may be public sector bodies, with pricing policies set by government or they may be purely private sector companies seeking to profit from the demand for spatial data. Public policy towards spatial data pricing differs greatly across different countries. In the US, the federal government regards spatial data as a public good. Basic spatial data should logically only be collected once, in the US it is seen as appropriate for the government to do this and then make this data available at little or no cost to the public. In contrast, government policy in Canada and in most European countries is to seek to recover some or all the cost of spatial data collection from those who use it [7]. However, where data is sold by public agencies, rather than given away, there is more incentive to reorganize that data to meet the needs of potential users. While basic spatial data is free in the US, there are many private sector data providers who add value to this basic data for a fee, making it more useful for an extended range of applications.

Other spatial data-sets may originate in organizations whose primary role is not spatial data collection. Such organizations collect data primarily for their own needs, but may make it available to others if there is an incentive for doing so. For example, an electricity supply utility will have GIS data on the location of its power lines.

The cost of building this data-set can be justified by the operational needs of the utility. Nevertheless, data on the availability of high-voltage power supplies might be useful information for a variety of planning and location analysis applications, so the data has value outside the organization which collected it. The range of possible applications for spatial data means that data will be used in combinations which were probably never envisaged by the originator of that data. Furthermore, the type of organization that collects spatial data for its own needs is not focused on providing spatial data to external users. Government policies might create an environment where utilities were expected to share data, but appropriate licensing and payment schemes are also needed to regulate this sharing.

Key Applications

Business decision makers are concerned with the business issues of their organization, so potential business users will want appropriate data sets in a convenient format to address these issues. For example, a business located in the city of Maastricht, in the Netherlands, may want to profile the potential customer base within one or two hour travel time from their premises. This will include customers in Germany, the Netherlands, Belgium and perhaps even Luxembourg. If this organization wants to buy spatial data for this application, they may find that the data for these different countries is not compatible, owing to the use of different technical approaches to the collection and display of spatial data. Fortunately, harmonization initiatives continue to address some of these technical compatibility problems, for an example in this region (see www.x-border-gdi.org). However, in addition to technical problems, business related problems also exist. For instance, can data for the precise area of relevance be purchased separately? In this situation only data-sets including an entire country or province may be available, even if most of the content in these data-sets is not needed for this specific application.

Digital spatial data can be readily copied, and subsequently distributed beyond those who have paid for it. Consequently, data-sets are usually made available with restrictions on their use. For example an organization may only be able to use data for its own needs, but are not permitted to pass the data on to third parties. The restrictions on spatial data use can be an obstacle to its widespread use, and the diversity of economic models used by public and private sector data providers fragments the market. This means that a Maastricht-based organization might find that the usage restrictions on their Belgian data are quite different from those relating to the data for the German part of their service area. Even within the same country a particular GIS application might require data from different original providers and similar problems may arise. For instance, road network data may be packaged for different regions and under different usage conditions than population data. As it can be electronically distributed, spatial data may be seen as a "*downloadable good*," analogous to existing virtual products in the software and music fields. The need to restrict the copying and further distribution of these products has lead to the development of techniques for Digital Rights Management (DRM), which offers the user a license granting them certain rights. A license is a digital data file that specifies certain usage rules for the digital content. Usage rules will restrict the frequency of access, expiration date, transfer to other devices and replication of data. There are two main approaches to restricting use, active approaches use technically based hardware or software based restrictions to attempt to enforce license restrictions, while passive approaches seek to impose high legal penalties for illicit use.

A comprehensive spatial data source (e. g. a geolibrary or geoportal) is likely to include a diverse range of data, under various licensing conditions. Spatial data are generally used by businesses, and are of most interest to organizations conducting business in the region the data relates to. An organization based in, or doing substantial business with, a particular country is quite likely to respect the laws of that country, and to respect restrictions on the use of spatial data. However, one important obstacle to compliance with licensing terms arises when users are unclear as to the exact restrictions and because there is no straightforward way to achieve compliance. A successful market for spatial data must have clear information on the restrictions applicable to that data and a convenient way for law abiding users to respect those restrictions. Projects such as the GeoData Commons Research Project at the University of Maine (www.spatial.maine.edu/geodatacommons) seek to clarify the licensing structure for spatial data [8]. These projects are largely based on the availability of free data; further work is needed on the full range of commercial licensing arrangements that are needed for a comprehensive market.

Future Directions

While the GIS software market is a large one, there are a relatively small number of major players. Where there is a dominant player in a market, then the file formats used by that player can become the *de facto* data standard. For example, documents are often exchanged using the *.doc* format used in Microsoft Word. In the spatial domain, the market is dominated by the ESRI Corp and the *.shp* and *.E00* file formats used by that company's products has become something of a de facto standard. These proprietary standards may be extended over time to include features which facilitate a spatial data market. However, if a proprietary format becomes dominant, then data provision can be greatly influenced by the dominant software vendor as well as the data providers, which may restrict the growth of the market. Data standards also originate in standards initiatives originating from collective organizations such as the GeoData Alliance (www.geoall.net). Such organizations work to facilitate open standards for the exchange of data. There are also initiatives to provide models relevant to the spatial data market; the Geospatial Digital Rights Management (GeoDRM) initiative of the Open Geospatial Consortium (OGC) (www.opengeospatial.org) is one example.

The spatial data market is likely to evolve to a situation where spatial software can connect electronically to spatial libraries to obtain spatial information. Data would be available in these libraries in a format containing embedded information on license restrictions. The software would allow use of this data in accordance with the license restrictions and would facilitate extension of a license as required. Digital watermarking could be included in spatial data-sets and used to trace data use. The spatial data market can evolve to take advantage of approaches developed in the e-commerce domain to encourage the sharing of spatial data, facilitating the use of spatial data in a wide range of potential applications.

Cross References

► Data Infrastructure, Spatial
► Information Services, Geography
► OGC's Open Standards for Geospatial Interoperability

Recommended Reading

1. Masser, I.: The future of spatial data infrastructures. ISPRS Workshop on Service and Application of Spatial Data Infrastructure, XXXVI (4/W6). Swets & Zeitlinger B.V, Hangzhou, China (2005)
2. FGDC: Framework, Introduction and GuideBook. Federal Geographic Data Committee, Washington (1997)
3. Bernard, L., et al.: Towards an SDI Research Agenda. 11 EC-GIS Workshop. Alghero, Sardinia (2005)
4. Maguire, D.J., Longley, P.A.: The emergence of geoportals and their role in spatial data infrastructures. Comput. Environ. Urban Syst. **29**(1), 3–14 (2005)
5. Goodchild, M.F.: The Geolibrary, Innovations in GIS V. In: Carver, S. (ed.), pp. 59–68. Taylor and Francis, London (1998)
6. Boxall, J.: Geolibraries, the Global Spatial Data Infrastructure and Digital Earth. Int. J. Spec. Libr. **36**(1), 1–21 (2002)
7. Craglia, M., Masser, I.: Access to Geographic Information: a European Perspective. J. Urban Regional Inf. Syst. Assoc. (URISA) **15**(APA I), 51–59 (2003)
8. Onsrud, H., et al.: Public Commons of Geographic Data: Research and Development Challenges. Geographic Information Science, Third International Conference, GIScience, Adelphi, MD, USA, 2004. Springer, Heidelberg (2004)

Market-Basket Analysis

► Frequent Itemset Discovery

Marketing Information System

► Decision-Making Effectiveness with GIS

Markov Random Field (MRF)

► Hurricane Wind Fields, Multivariate Modeling

Mash-Ups

► Information Services, Geography

Massive Evacuations

► Emergency Evacuation Plan Maintenance

Mathematical Foundations of GIS

TIMOTHY G. FEEMAN
Department of Mathematical Sciences,
Villanova University, Villanova, PA, USA

Definition

- *Geodesy* is the branch of mathematics concerned with the shape and area of the earth and with the location of points on it.
- *Cartography* is the art, science, and practice of making maps.
- A *map projection*, or simply a projection, is any systematic representation of the earth's surface onto another surface.
- The *Global Positioning System*, or GPS, comprises a network of satellites that orbit the earth and, by radio communications with land-based receivers, enable the accurate determination of the coordinates of points on the earth's surface.
- *Spherical geometry* is the study of lines, angles, and areas on a spherical surface.

M

Historical Background

The foundation of geographical information science (GIS) lies in our ability to determine the size and shape of the earth, locate points on its surface, measure its features, and to portray the earth in maps. Thus, geodesy and cartography form the basis of GIS. In turn, both of these subjects are built on strong mathematical foundations.

The Shape and Size of the Earth

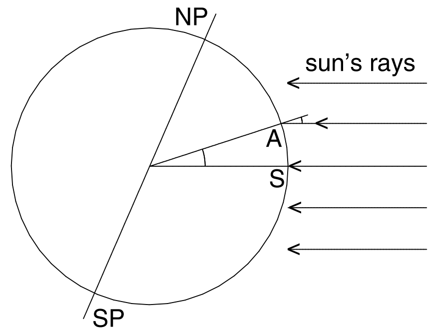
In this age of space exploration, photographs of Earth and other heavenly bodies taken from space offer convincing evidence that Earth's basic shape is spherical. This conception of a spherical Earth has endured, though not without some lapses, at least since the sixth century B.C., when Anaximander and Thales of Miletus, two of the earliest classical Greek geometers, described the earth as a sphere positioned at the center of a huge celestial sphere to which were fixed the other visible planets and stars.

Towards the end of the seventeenth century, Isaac Newton's work on gravitation and planetary motion led him to conclude that Earth had the shape of an ellipsoid, flattened at the poles and somewhat bulging around the equator. Newton's conjecture was confirmed by French-sponsored expeditions to Peru and Lapland during the 1730s in which arcs of meridians at high and low latitudes were measured. In the early 1800s, Gauss and others provided further verifications. Since the introduction of satellite technology, new measurements have resulted in the development of several reference ellipsoids, including the World Geodetic Systems ellipsoid of 1984 (WGS 84, for short), developed by the US Defense Mapping Agency, and the Geodetic Reference System ellipsoid of 1980 (GRS 80), adopted by the International Union of Geodesy and Geophysics in 1979.

When the highest level of precision is needed, an ellipsoid provides the best mathematical model for the earth's shape. In this article, for simplicity's sake, a spherical Earth is assumed, with measurements and calculations made accordingly.

Assuming the earth to be a sphere, the single most important measurement is its circumference, which was estimated by Eratosthenes of Alexandria in roughly 230 B.C.

Eratosthenes knew that, at noon on any given day of the year, the angle of the sun above the horizon would be different at two different places located north and south of one another. Assuming the earth to be a sphere and all of the sun's rays to be parallel, Eratosthenes called upon a basic geometric fact – that a line transversal to two parallel lines will make equal angles with both – to conclude that the difference between the angles of the sun would correspond to the central angle of the portion of the earth's circumference between the two points (Fig. 1). He also knew that, at noon on the summer solstice, the sun shone directly overhead in the town of Syene (now Aswan, Egypt), famously illuminating a well there. Eratosthenes then determined that, on the summer solstice, the noonday sun in Alexandria was one-fiftieth part of a full circle (7.2°) shy of being directly overhead. It followed that the distance between Syene and Alexandria must be one-fiftieth part of a full circumference of the earth. The distance between the two towns was measured to be about 5000 stadia (approximately 500 miles). Multiplication by 50 yielded Eratosthenes' remarkably accurate estimate of 250000 stadia, about 25000 miles, for the circumference of the earth. The basic method employed by Eratosthenes is completely valid and is still used today.



Mathematical Foundations of GIS, Figure 1 The sun is overhead at Syene (*S*). The angle of the sun at Alexandria (*A*) is the same as the central angle

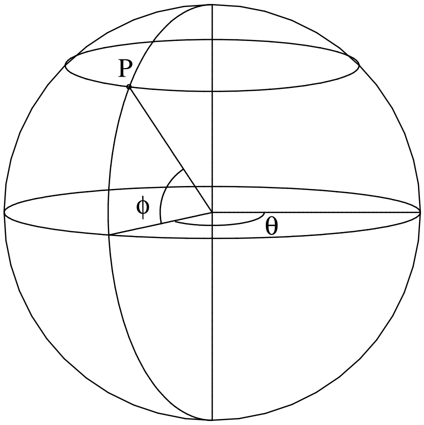
In fact, the equatorial and polar radii and circumferences of the earth are different, because of the earth's ellipsoidal shape. The mean radius of the earth is approximately 6371 kilometers.

Scientific Fundamentals

Location

The latitude of any given point is defined to be the difference between the angles made by the sun at noon at the point in question and at the equator. This is illustrated in Fig. 2. Latitude is designated as north (N) or south (S) according to which hemisphere the point lies in. The points at the same latitude form a circle whose plane is parallel to that of the equator. Thus, a circle of latitude is called a *parallel*.

As the earth rotates about its axis, the position of the sun in the sky changes, ascending from the eastern horizon at dawn to its zenith at noon, then descending until it sets in the west. The arrival of local solar noon, the moment at which the sun reaches its zenith, is a simultaneous event at



Mathematical Foundations of GIS, Figure 2 The point P has longitude θ and latitude ϕ

all points along a semi-circular arc, called a *meridian*, that extends from the north pole to the south pole. Where two meridians come together at the poles, they form an angle that is the basis for determining *longitude*. The difference in longitudes of two locations is the portion of a full circle through which the earth rotates between the occurrences of local solar noon at the two places. Thus, the measurement of longitude is fundamentally a problem of the measurement of time.

One way to measure time differences between two places is by the relative positions in the sky of various celestial bodies, such as certain stars, the planets, or the moons of Jupiter. The astrolabe and the sextant were among the tools developed to measure these positions, and elaborate tables were compiled showing known positions of celestial objects. Alternatively, longitude can be determined using two clocks, one set to the time at a fixed location and the other to local time. Though sufficiently accurate clocks are readily available today, their introduction just a few centuries ago marked a giant leap in the technology of navigation. The first sea-worthy chronometer was developed in England by John Harrison and presented to the English Longitude Board in 1735, though it took some years of refinements in size, weight, and ease of reproduction for the new clocks to catch on.

In addition to the technical problem of measuring time differences, there is the political issue of deciding which meridian will serve as the reference, or prime meridian, for longitude calculations. Ptolemy placed his prime meridian through the Canary Islands. Others have used the meridians through Mecca, Jerusalem, Paris, Rome, Copenhagen, the Cape Verde Islands, St. Petersburg, Philadelphia, and more. After 1767, when the Royal Observatory in Greenwich, England, published the most comprehensive tables of lunar positions available, sailors increasingly calculated their longitude from Greenwich. This practice became official in 1884 when the International Meridian Conference established the prime meridian at Greenwich. Longitude is designated as east (E) or west (W) according to whether local noon occurs before or after local noon in Greenwich. Latitude and longitude together give a complete system for locating points on the earth's surface.

Coordinates

Where cartographers often measure angles in degrees, here **angles will be measured in radians** in order to simplify trigonometric computations. Also, positive and negative angle measurements will be used instead of the directional designations north/south or east/west, with south and west assigned negative values. The symbols θ and φ will denote longitude and latitude, respectively. Thus, the point with longitude 75° W and latitude 40° N has coordinates ($\theta = -75\pi/180$, $\phi = 40\pi/180$) while the point with longitude 75° E and latitude 40° S has coordinates ($\theta = 75\pi/180$, $\phi = -40\pi/180$).

For the Cartesian coordinate system in three-dimensional space, the line through the north and south poles will be taken as the z-axis, with the north pole on the positive branch. The plane of the equator corresponds to the xy-plane with the positive x-axis meeting the equator at the prime meridian and the positive y-axis meeting the equator at the point with longitude $\pi/2$ (or 90° E). On a sphere of radius R, then, the point with longitude θ and latitude φ will have Cartesian coordinates $(x, y, z) = (R\cos(\theta)\cos(\phi), R\sin(\theta)\cos(\phi), R\sin(\phi))$. Conversely, latitude and longitude can be recovered from the Cartesian coordinates. The equation $\phi = \arcsin(z/R)$ determines ϕ uniquely as an angle between $-\pi/2$ and $\pi/2$. Also, θ satisfies the equations $\tan(\theta) = y/x$, $\cos(\theta) = x/\sqrt{x^2+y^2}$, and $\sin(\theta) = y/\sqrt{x^2+y^2}$. Any two of these together will determine a unique angle θ between $-\pi$ and π.

Distance

The distance between two points is the length of the shortest path connecting the two points. On a sphere, the path must lie entirely on the surface and, so, cannot be a straight line segment as it is in a plane. Instead, the shortest path is the straightest possible one which, intuitively, is an arc of the largest possible circle, called a great circle.

A *great circle* on a sphere is the intersection of the sphere with a plane that contains the center of the sphere. Every great circle has a radius equal to that of the sphere. The equator is a great circle while each meridian is half of a great circle. Any two great circles must intersect each

M

other at a pair of antipodal points. Indeed, the planes defined by the circles will intersect in a line through the sphere's center, which, therefore, will intersect the sphere at two opposite points.

Any two non-antipodal points on the sphere determine a unique great circle, namely the intersection of the sphere with the plane generated by the two points together with the center of the sphere. The two points divide this circle into two arcs, the shorter of which is the shortest path connecting the points. If O denotes the center of a sphere of radius R and by A and B the two points of interest, then the distance between A and B along the shorter great circle arc is equal to the product of R and the central angle, measured in radians, formed by the two vectors $\overrightarrow{OA}$ and $\overrightarrow{OB}$. (This is essentially the definition of radian measure.)

To derive a formula for distance in terms of the longitudes and latitudes of the points in question, assume for simplicity's sake that the sphere has radius $R = 1$ unit and let the points A and B have longitudes θ_1 and θ_2 and latitudes ϕ_1 and ϕ_2, respectively. Converting to three-dimensional Cartesian coordinates,

$$\begin{aligned} A &= (\cos\phi_1 \ \cos\theta_1,\ \cos\phi_1 \sin\theta_1,\ \sin\phi_1) \ \text{ and} \\ B &= (\cos\phi_2 \ \cos\theta_2,\ \cos\phi_2 \sin\theta_2,\ \sin\phi_2)\,. \end{aligned}$$

From vector geometry, the *cosine* of the angle between the two vectors $\overrightarrow{OA}$ and $\overrightarrow{OB}$ is given by their dot product divided by the product of the vectors' lengths, each of which is 1 in this case. The value of the dot product is

$$\begin{aligned} &\overrightarrow{OA} \bullet \overrightarrow{OB} \\ &\quad = \cos\phi_1 \cos\theta_1 \cos\phi_2 \cos\theta_2 + \cos\phi_1 \sin\theta_1 \cos\phi_2 \sin\theta_2 \\ &\qquad + \sin\phi_1 \sin\phi_2 \\ &\quad = \cos\phi_1 \ \cos\phi_2 \,(\cos\theta_1 \ \cos\theta_2 + \sin\theta_1 \ \sin\theta_2) \\ &\qquad + \sin\phi_1 \ \sin\phi_2 \\ &\qquad = \cos\phi_1 \ \cos\phi_2 \ \cos(\theta_1 - \theta_2) + \sin\phi_1 \ \sin\phi_2\,. \end{aligned}$$

The angle between the vectors is then

$$\begin{aligned} \text{angle} &= \arccos(\overrightarrow{OA} \bullet \overrightarrow{OB}) \\ &= \arccos(\cos\phi_1 \ \cos\phi_2 \ \cos(\theta_1 - \theta_2) \\ &\quad + \sin\phi_1 \ \sin\phi_2)\,. \end{aligned} \tag{1}$$

Now multiply this angle by the radius of the sphere to get the distance. That is,

$$\begin{aligned} \text{distance} = R \arccos(&\cos\phi_1 \ \cos\phi_2 \ \cos(\theta_1 - \theta_2) \\ &+ \sin\phi_1 \ \sin\phi_2)\,. \end{aligned} \tag{2}$$

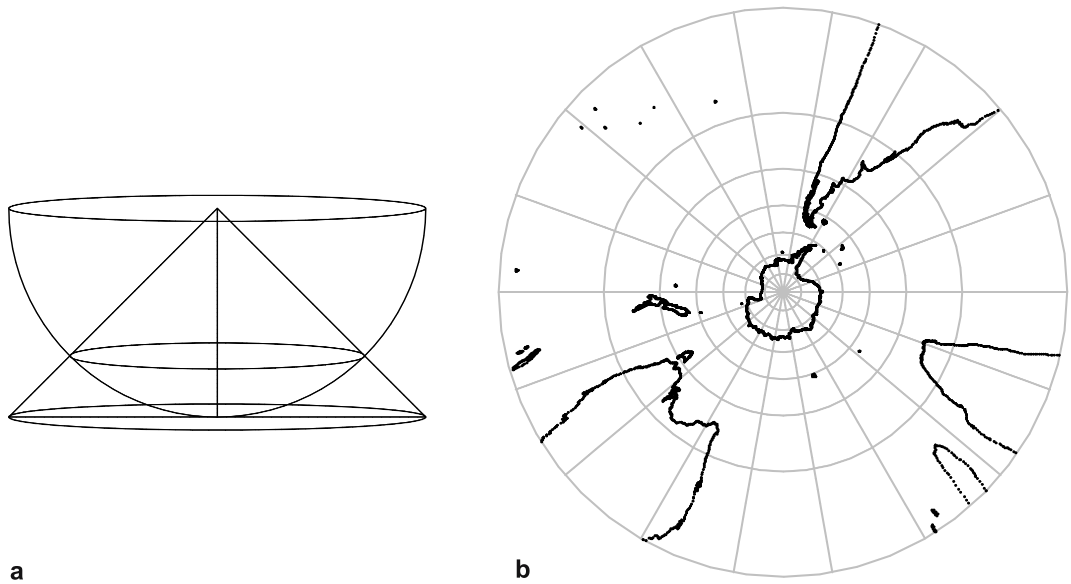
For example, London, England, has longitude $\theta_1 = 0$ and latitude $\phi_1 = 51.5\pi/180$, while $\theta_2 = 116.35\pi/180$ and $\phi_2 = 2\pi/9$ are the coordinates of Beijing. Therefore, the central angle between London and Beijing is about 1.275 radians. The mean radius of the earth is about $R = 6371$ kilometers and, hence, the great circle distance from London to Beijing is approximately $1.275R$, or 8123 kilometers. The great circle route is illustrated in Fig. 3.

Mathematical Foundations of GIS, Figure 3 Great circle route from London to Beijing

The most important map projection for the depiction of great circle routes is the gnomonic projection, which is constructed by projecting a spherical globe onto a plane tangent to the globe using a light source located at the globe's center. Any two points on the sphere, along with the light source for the projection, define a plane that intersects the sphere in a great circle and the plane of the map in a straight line, which is, therefore, the image of the great circle joining the points. In other words, the gnomonic projection has the property that the shortest route connecting any two points A and B on the sphere is projected onto the shortest route connecting their images on the flat map.

The gnomonic projection was most likely known to Thales of Miletus and would have been particularly useful to navigators and traders of the Golden Age of Greece, a time of intensified trade and geographical discovery during which the Bronze Age gave over to the Age of Iron.

The gnomonic projection can be constructed from elementary geometry. Place a globe of radius R on a flat piece of paper with the south pole at the bottom and with a projecting light source at the center of the globe. Points on or above the equator won't project onto the paper, so the map will show only the southern hemisphere. Arrange the map's coordinate axes so that the prime meridian is projected onto the positive x-axis, in which case the image of the meridian at longitude θ makes an angle of θ, measured counter-clockwise, with the positive x-axis. The parallels, meanwhile, will be shown on the map as concentric circles having the pole as their common center. When the globe is



Mathematical Foundations of GIS, Figure 4 The gnomonic projection. The basic construction is depicted (**a**), and the resulting map of most of the southern hemisphere (**b**)

viewed from the side, as in Fig. 4, two similar right triangles can be seen. Each has the center of the globe as a vertex. The horizontal side of the smaller triangle is a radius of the parallel at latitude ϕ. Hence, the vertical and horizontal sides of the smaller triangle have lengths $R\cos(\pi/2+\phi)$ and $R\sin(\pi/2+\phi)$, respectively. (Note that $\phi < 0$ in this context.) The vertical side of the larger triangle is a radius of the globe, so its length is R. Let $r(\phi)$ denote the length of the horizontal side of the larger triangle, which is the radius of the projected image of the parallel at ϕ. The proportionality of sides for similar triangles yields the equation

$$\begin{aligned}\frac{r(\phi)}{R} &= \frac{R\sin(\pi/2+\phi)}{R\cos(\pi/2+\phi)}, \quad \text{whence}\\ r(\phi) &= \frac{R^2\sin(\pi/2+\phi)}{R\cos(\pi/2+\phi)} = R\tan(\pi/2+\phi)\\ &= -R\cot(\phi)\,.\end{aligned} \tag{3}$$

A base grid for a gnomonic projection of most of the southern hemisphere can now be constructed, either on a computer or by hand using a protractor, ruler, and compass. The resulting map is shown in Fig. 4.

Spherical Triangles

A spherical triangle is formed when arcs of three different great circles meet in pairs. For a planar triangle, the sum of the three interior angles is always π radians. For a spherical triangle, however, the sum of the interior angles is directly related to the size of the triangle. For instance, two points on the equator and a third point near the equator define a spherical triangle that almost fills up a hemisphere. Each angle will be almost π, so the sum of the angles will be just under 3π. On the other hand, a small triangle will be nearly flat, so its angles will add up to a number close to π. The exact relationship between the area of a spherical triangle and the sum of its angles is expressed in the formula

$$\text{spherical triangle area} = R^2\ (\text{sum of the angles} - \pi)\,. \tag{4}$$

To prove formula (4), consider a spherical triangle with corners at A, B, and C, where, with no loss of generality, the edge $\widehat{AB}$ lies on the horizon with A at the north pole and C is in the front hemisphere. Extend the arcs $\widehat{AC}$ and $\widehat{BC}$ to divide the hemisphere into four parts, as depicted in Fig. 5, with areas labeled as a, b, c, and t. The area of the triangle is t.

When the arcs $\widehat{AB}$ and $\widehat{AC}$ are extended into semicircles, they intersect at the antipode to A and enclose a portion of the sphere's surface, called a *lune*, whose area is $a+t = (\angle BAC/\pi)(2\pi R^2) = 2(\angle BAC)R^2$. Similarly, one can form two other lunes with areas $b+t = 2(\angle ABC)R^2$ and $c+t = 2(\angle BCA)R^2$. Note that the lune formed when the arcs $\widehat{CA}$ and $\widehat{CB}$ are extended actually consists of the region t together with a copy of region c on the back side of the sphere. Adding up these three areas, one has $a+b+c+3t = 2R^2$ (sum of the angles). But regions a, b, c, and t collectively form a hemisphere, so, also, $a+b+$

M

$c + t = 2\pi R^2$. Subtract this from the previous equation to get $2t = 2R^2(\text{sum of the angles}) - 2\pi R^2$, from which it follows that $t = R^2$ (sum of the angles $- \pi$), as was to be proved.

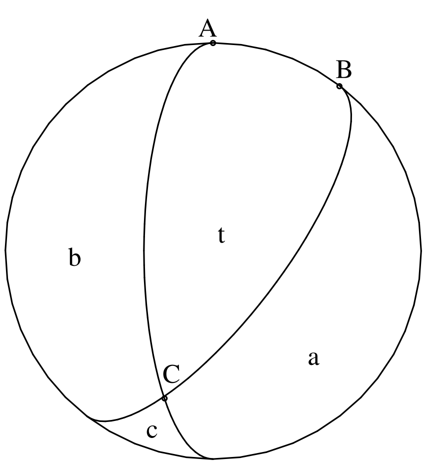
Classical Spherical Trigonometry

Two classical results from spherical trigonometry that have useful applications to problems such as measurement of distance and determination of azimuths are the Law of Cosines and the Law of Sines. Like their Euclidean counterparts, they relate the measurements of various parts of a triangle.

Referring again to the spherical triangle in Fig. 5, note that each side of the triangle, being an arc of a great circle, has length equal to R times the central angle formed by the vectors connecting the origin to the corresponding vertices. With this in mind, let α, β, and γ denote the central angles corresponding to the sides opposite the vertices A, B, and C, respectively. Thus, $\alpha := \widehat{BC}/R$, $\beta := \widehat{AC}/R$, and $\gamma := \widehat{AB}/R$.

Without loss of generality, assume that A is at the north pole, so that its latitude is $\phi_1 = \pi/2$ and its longitude θ_1 is arbitrary. The points B and C have generic coordinates $B(\theta_2, \phi_2)$ and $C(\theta_3, \phi_3)$. An application of formula (1) above yields

$$\begin{aligned}\cos(\alpha) &= \cos(\phi_2)\cos(\phi_3)\cos(\theta_2 - \theta_3)\\ &\quad + \sin(\phi_2)\sin(\phi_3)\,,\\ \cos(\beta) &= \sin(\phi_3) \;\text{ since } \phi_1 = \pi/2, \text{ and}\\ \cos(\gamma) &= \sin(\phi_2) \;\text{ since } \phi_1 = \pi/2\,.\end{aligned}$$



Mathematical Foundations of GIS, Figure 5 The spherical triangle with vertices at A, B and C has area t. The areas a, b, c, and t together fill up a hemisphere. Also, each of the areas a, b, and c forms a lune when combined with t

Moreover, the angle in the triangle itself at vertex A is $\angle BAC = (\theta_2 - \theta_3)$, the difference in the longitudes of B and C.

It follows that $\sin(\beta) = \sqrt{1 - \cos^2(\beta)} = \cos(\phi_3)$ and that $\sin(\gamma) = \sqrt{1 - \cos^2(\gamma)} = \cos(\phi_2)$. Hence,

$$\cos(\alpha) = \cos(\beta)\cos(\gamma) + \sin(\beta)\sin(\gamma)\cos(\angle BAC)\,. \tag{5}$$

This is the Law of Cosines for spherical trigonometry.

As an example, consider the spherical triangle formed by the north pole (A), Beijing (B), and London (C). As was shown above, $\alpha = \widehat{BC}/R \approx 1.275$ radians. Moreover, from the latitudes of Beijing and London, one has $\beta = \pi/2 - 51.5\pi/180 \approx 0.672$ radians and $\gamma = \pi/2 - 2\pi/9 \approx 0.873$ radians. The Law of Cosines (5) implies that

$$\cos(\gamma) = \cos(\alpha)\cos(\beta) + \sin(\alpha)\sin(\beta)\cos(\angle ACB)\,.$$

The vertex angle at London is, therefore,

$$\begin{aligned}\angle ACB &= \arccos\left(\frac{\cos(\gamma) - \cos(\alpha)\cos(\beta)}{\sin(\alpha)\sin(\beta)}\right)\\ &\approx 0.8005 \text{ radians or } 45.866^\circ\,.\end{aligned}$$

Similarly, the vertex angle at Beijing is $\angle ABC \approx 0.6227$ radians, or 35.678°. As the angle at the north pole is $\angle BAC = 116.35\pi/180 - 0 \approx 2.031$ radians, it follows from formula (4) that the area of this spherical triangle is approximately $(6371)^2(2.031 + .8005 + .6227 - \pi) \approx 12,688,727$ square kilometers. This is not quite 2.5% of the earth's surface.

The Law of Cosines (5) implies that

$$\cos(\angle BAC) = \frac{\cos(\alpha) - \cos(\beta)\cos(\gamma)}{\sin(\beta)\sin(\gamma)}\,.$$

Hence, using basic trigonometric identities, it follows that

$$\frac{\sin^2(\angle BAC)}{\sin^2(\alpha)} = \frac{1 - \cos^2(\alpha) - \cos^2(\beta) - \cos^2(\gamma) + 2\cos(\alpha)\cos(\beta)\cos(\gamma)}{\sin^2(\alpha)\sin^2(\beta)\sin^2(\gamma)}\,.$$

This last expression is symmetric in α, β, and γ and, therefore, the value of the left-hand side does not depend on the vertex chosen. That is,

$$\frac{\sin(\angle BAC)}{\sin(\alpha)} = \frac{\sin(\angle ABC)}{\sin(\beta)} = \frac{\sin(\angle ACB)}{\sin(\gamma)}\,, \tag{6}$$

which is the Law of Sines for spherical trigonometry.

Returning to the example of the triangle formed by the north pole, London, and Beijing, it has been shown already that $\angle BAC \approx 2.031$ radians, $\alpha \approx 1.275$ radians, and $\gamma = \pi/2 - \phi_2 \approx 0.873$ radians. Thus, by the Law of Sines (6),

$$\angle ACB = \arcsin\left(\frac{\sin(\angle BAC)\sin(\gamma)}{\sin(\alpha)}\right) \approx 0.8005 \text{ radians or } 45.866^\circ .$$

Key Applications

The Global Positioning System

Perhaps the single most important aspect of mapping is the question of location. Indeed, a substantial portion of Ptolemy's classic treatise, *Geography*, which contains several of his most famous maps, is taken up with lists of the longitude and latitude coordinates of various places, much of this information having been gleaned from accounts of travelers. The latest, and to date the most accurate, method for determining the coordinates of a point on the earth is the Global Positioning System, or GPS.

Originally introduced by the United States Department of Defense in the 1980s, and made fully functional in 1995, the GPS consists of twenty-four solar-powered satellites that orbit the earth in nearly circular half-day orbits approximately 20,200 kilometers above the earth's surface. There are four satellites in each of six distinct orbital planes with each plane inclined at an angle of 55° from the plane of the equator. The overall arrangement ensures that every point on earth is nearly always visible from at least four satellites. Each satellite is equipped with a highly accurate atomic clock and continuously transmits, via a microwave radio, the exact time of its internal clock, its precise position, and other secondary information.

A GPS receiver, located somewhere on the earth, can compute its position, as well as the exact time, by determining its distances from four of the satellites. The distance to each satellite is calculated by measuring the time delay between the transmission and the reception of the satellite's signal. In principle, this information locates the receiver at the intersection of four spheres, one centered at each of the four satellites. Since the receiver's position has only three coordinates in space, it would seem to suffice to use three satellites. However, errors in the receiver's clock should be treated as a fourth variable in the equations, thus necessitating the fourth satellite. With only three satellites, it is still possible to obtain a location at sea level.

The key to getting an accurate position is to measure the time delays accurately. The effects of the atmosphere on the propagation of the radio signals are significant in this regard. These effects depend on the transmission frequency, so one solution is to have the satellites broadcast simultaneously on two different frequencies. Using the difference in the time delays in the reception of the two signals, the GPS receiver can adjust for the atmospheric effects. Alternatively, since the effects will be similar for an entire region, a ground station can communicate an appropriate correction factor to GPS receivers in its area. Other factors that influence the transit time of the signals include drifts in the accuracy of the atomic clocks due to aging, radiation in space, power supply fluctuations, and even relativistic effects. For all of these factors, the cumulative effect is known for each satellite and is actually transmitted as part of the signal. Thus, the GPS receiver on the ground is able to account for these factors in its calculations.

So, for each value of the index $i = 1, 2, 3,$ and 4, let t_i denote the adjusted measurement by the GPS receiver of the time delay in the reception of the signal from satellite number i. If the receiver's clock were synchronized with those of the satellites, then the distance to satellite i would be $c \cdot t_i$, where c is the speed of light. But, if the receiver's clock were off by b seconds, then even a tiny value of b would result in huge errors in distance measurements. Of course, the error in the receiver clock is not known ahead of time, thus it remains a variable in the calculations. In short, the receiver computes the distance to satellite i as $c(t_i + b)$.

Let (x_i, y_i, z_i) denote the position in space of the ith satellite at the instant it transmits its signal. Then the receiver is located at the point (x, y, z) whose coordinates satisfy the equations

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 - c^2(t_i + b)^2 = 0 \quad (7)$$

for $i = 1, 2, 3,$ and 4. This system can be solved, to any desired degree of accuracy, by standard numerical methods such as least squares. Generally, at least ten digits must be computed. The computed value of b allows the GPS receiver to adjust its internal clock accordingly, though future computations will still assume the clock to be in error. The longitude and latitude of the receiver can be recovered from the Cartesian coordinates x, y, and z, as discussed earlier, while the elevation is the difference between $\sqrt{x^2 + y^2 + z^2}$ and sea level at the point (θ, ϕ).

To see how closely this procedure estimates the location of the GPS receiver, observe that, in the system (7), each of the unknowns x, y, and z will vary if the values of the t_i are allowed to vary. For instance, if x is viewed as a function of the t_i s, then, according to multivariable linear approximation, the differential dx is given by

$$\mathrm{d}x = \frac{\partial x}{\partial t_1}\mathrm{d}t_1 + \frac{\partial x}{\partial t_2}\mathrm{d}t_2 + \frac{\partial x}{\partial t_3}\mathrm{d}t_3 + \frac{\partial x}{\partial t_4}\mathrm{d}t_4 . \quad (8)$$

M

Similar formulas prevail for y and z. If $|\mathrm{d}t_i| < M$ for all i, then it follows that

$$|\mathrm{d}x| < q\left(\left|\frac{\partial x}{\partial t_1}\right| + \left|\frac{\partial x}{\partial t_2}\right| + \left|\frac{\partial x}{\partial t_3}\right| + \left|\frac{\partial x}{\partial t_4}\right|\right) M\,.$$

In a typical real-life scenario, this yields $|\mathrm{d}x| < q(3 \cdot 10^9)\,M$. Thus, if the values of all t_i are accurate to within 10^{-9} seconds (a nanosecond), then the estimated value of x will be within 3 meters of its correct value. If the accuracy of the t_i is only $3 \cdot 10^{-8}$ seconds, then a typical measurement of x will be within 90 meters of the actual value.

The estimate of $|\mathrm{d}x|$ just discussed requires that the values of the partial derivatives $\partial x/\partial t_i$ at the solution point are known. To determine these values, first solve the system (7) using the given satellite data. Next, take the partial derivative with respect to t_1, say, on both sides of every equation in (7), treating x, y, z, and b as functions of t_1 and treating the x_i, y_i, and z_i as constants. This yields a system of four linear equations in the unknowns $\partial x/\partial t_1$, $\partial y/\partial t_1$, $\partial z/\partial t_1$, and $\partial b/\partial t_1$. Similar systems obtained by differentiating (7) with respect to t_2, t_3, and t_4 lead to the following matrix equation.

$$\begin{bmatrix} 2(x-x_1) & 2(y-y_1) & 2(z-z_1) & -2c^2(t_1+b) \\ 2(x-x_2) & 2(y-y_2) & 2(z-z_2) & -2c^2(t_2+b) \\ 2(x-x_3) & 2(y-y_3) & 2(z-z_3) & -2c^2(t_3+b) \\ 2(x-x_4) & 2(y-y_4) & 2(z-z_4) & -2c^2(t_4+b) \end{bmatrix}$$
$$\begin{bmatrix} \partial x/\partial t_1 & \partial x/\partial t_2 & \partial x/\partial t_3 & \partial x/\partial t_4 \\ \partial y/\partial t_1 & \partial y/\partial t_2 & \partial y/\partial t_3 & \partial y/\partial t_4 \\ \partial z/\partial t_1 & \partial z/\partial t_2 & \partial z/\partial t_3 & \partial z/\partial t_4 \\ \partial b/\partial t_1 & \partial b/\partial t_2 & \partial b/\partial t_3 & \partial b/\partial t_4 \end{bmatrix}$$
$$= \begin{bmatrix} 2c^2(t_1+b) & 0 & 0 & 0 \\ 0 & 2c^2(t_2+b) & 0 & 0 \\ 0 & 0 & 2c^2(t_3+b) & 0 \\ 0 & 0 & 0 & 2c^2(t_4+b) \end{bmatrix}. \tag{9}$$

Once the coefficient matrix on the left-hand side and the matrix on the right-hand side of (9) have been evaluated at the solution point, then the values of all of the partial derivatives can be determined by multiplying both sides of (9) by the inverse of the coefficient matrix. Knowing these values, as well as the maximum error M in the accuracy of the t_i, one can then estimate the differentials $|\mathrm{d}x|$, $|\mathrm{d}y|$, and $|\mathrm{d}z|$ (and $|\mathrm{d}b|$, for that matter), and thereby estimate the accuracy of the location calculated by the receiver.

Map Projections

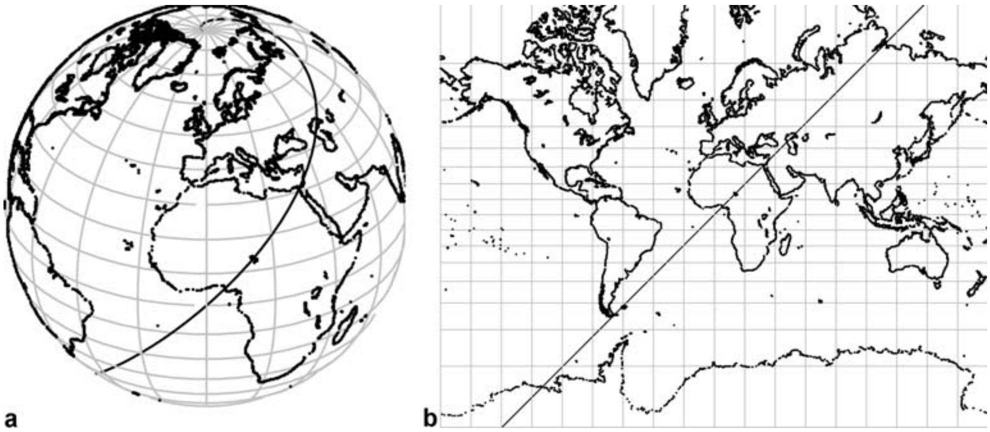
Ptolemy's goal, in *Geography*, was not only to collect the locations of as many places as possible, but to present them in the larger context of a portrait of the earth – a map. Maps go beyond mere location to reveal the many relationships that exist between different peoples and their environments. A host of map projections – specific formats for representing the earth or various parts of it – have been created over the millenia. While some projections are essentially artistic, many are designed on mathematical foundations with certain uses in mind. Of particular interest in GIS are maps designed for navigational purposes and those that are amenable to the display of statistical information.

Navigation and Mercator's Map

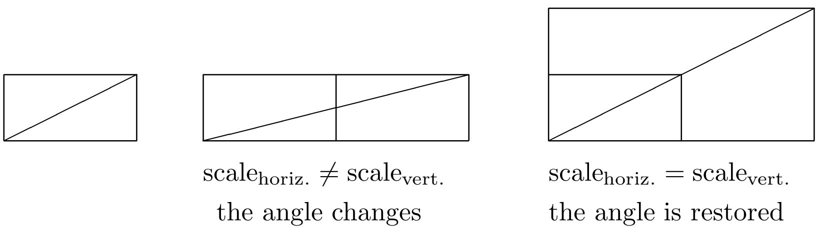
The gnomonic projection, discussed above, preserves shortest routes, and, thus, enables navigators to plot shortest routes quite easily, provided the points are not too far apart. However, to follow a great circle path generally requires continual changes in compass bearing, which is inconvenient. A more practical approach might be to plot a route that approximates the shortest one but requires only periodic changes in compass bearings. Hence, a map on which paths of constant compass bearing on the sphere were shown as straight lines would be a useful navigational tool. It was just such a map that the Flemish geographer Gerhard Kremer, better known as Mercator, presented in 1569 with the title *Nova et aucta orbis terrae descriptio ad usum navigantium emendate accommodata* (A new and enlarged description of the earth with corrections for use in navigation).

When following a path along the surface of the Earth, one's compass bearing at any given point is represented by the angle between the direction of the path and the meridian through that particular point. Thus, a path of constant compass bearing, called a *loxodrome*, makes the same angle with every meridian it crosses. A loxodrome generally appears as a spiral converging to one of the poles, as illustrated in Fig. 6. Mercator's problem was to figure out how to show all such spirals as straight lines on a map.

To solve this problem, consider first that all parallels and all meridians have constant compass bearings and, so, must be shown as straight lines on the map. Moreover, because the east-west direction is perpendicular to the north-south direction, the images of the parallels should be perpendicular to the images of the meridians. Thus, Mercator chose for the form of his map a rectangular grid in which all parallels of latitude are shown as horizontal lines and the meridians are equally spaced vertical lines. For simplicity, place the equator along the x-axis of a two-dimension-



Mathematical Foundations of GIS, Figure 6 A northeast/southwest loxodrome; Mercator's map for latitudes $-4\pi/9 < q\phi < q4\pi/9$



Mathematical Foundations of GIS, Figure 7 How scale factors affect angles

al Cartesian coordinate system and the meridian at longitude θ along the vertical line $x = \theta$, for $-\pi < q\theta < q\pi$. Thus, the overall width of the map will be 2π. The parallel at latitude ϕ will be shown as a horizontal line segment at height $y = h(\phi)$, where the function h is to be determined. A loxodrome on the globe that makes an angle of α with every meridian it crosses should be shown on the map as a straight line that makes an angle of α with every vertical line it crosses. As Fig. 7 illustrates, this goal will be achieved if, at each point on the map, the scale factor along the parallel, represented by the horizontal edge in the figure, is equal to the scale factor along the meridian, represented by the vertical edge in the figure.

On a reference globe of radius 1 unit, the parallel at latitude ϕ has a circumference of $2\pi\cos(\phi)$, while its image on Mercator's map has length 2π. Since the meridians are evenly spaced, every section of the parallel is stretched by the same amount. Hence, the scale factor of the map along the parallel at latitude ϕ is $M_p(\phi) = \sec(\phi)$.

On the same reference globe of radius 1, the arc of any meridian lying between latitudes ϕ and $(\phi + t)$ has length t while its image on the map has length $h(\phi + t) - h(\phi)$, the gap between the horizontal lines corresponding to the two parallels. Thus, the ratio between the map measurement and the globe measurement is $(h(\phi+t)-h(\phi))/t$. To obtain the exact value of the scale factor, let t approach 0. That is, the scale factor along any meridian at a point at latitude ϕ is given by

$$M_m(\phi) = \lim_{t\to 0} \frac{h(\phi + t) - h(\phi)}{t} = h'(\phi)\,,$$

the derivative of the height function for the parallels.

The solution to Mercator's problem, therefore, is to choose the height function $h(\phi)$ so that $M_m(\phi) = M_p(\phi)$. That is, $h'(\phi) = \sec(\phi)$. Also, $h(0) = 0$ since the equator lies on the x-axis. Together, these conditions imply that

$$h(\phi) = \int_0^{\phi} \sec(t)\,\mathrm{d}t = \ln|sec(\phi) + \tan(\phi)|\;. \tag{10}$$

Notice that, as the latitude gets close to $\pm\pi/2$, the scale factor $\sec(\phi)$ tends to infinity. This explains why Mercator's map shows regions in the northern latitudes to be so large compared to equatorial areas.

Equipped with both a gnomonic map and Mercator's map, a navigator can plot a useful route as follows. On the gnomonic map, draw the straight line connecting the starting and ending points. This represents the shortest possible route between the two points. Next, mark some convenient reference points along this route and locate these same reference points on the Mercator map. Now, on the Mercator map, connect the reference points with straight line segments. This is the actual travel route. It follows a constant compass bearing from one reference point to the next while staying reasonably close to the shortest route.

In general, a map projection with the property that the projected images of any two intersecting paths intersect at an angle equal to that between the two paths themselves is said to be *conformal*. Loosely, a conformal map is said to "preserve angles". Thus, Mercator's map is conformal.

An Equidistant Projection

The gnomonic projection is one of the classical projections handed down from ancient Greek geometry. Two others, the sterographic and orthographic projections, are constructed in a similar fashion with the projecting light source located, respectively, at the antipode of the point of tangency of the sphere and the paper and at infinity. A fourth classical projection, the azimuthal equidistant projection, is a strictly mathematical construction that does not utilize a light source. The term *azimuthal* is used to describe any projection that has a central point rather than a central line or lines. All azimuthal maps show great circle paths through the central point as straight lines. Appropriate spacing can also ensure that the distances along these great circle paths are shown in their correct proportions. The resulting map, commonly used for atlas maps of the polar regions, is called an *azimuthal equidistant projection.* For example, on an azimuthal equidistant map with the north pole as its central point, the meridians, being great circle arcs through the pole, are shown on the map as straight lines radiating out from the center. If the prime meridian is mapped onto the positive x-axis, then the image of the meridian at longitude θ, where $-\pi < q\theta < q\pi$, will make an angle of θ with the positive x-axis. As for the parallels, note that all points at latitude ϕ, where $-\pi/2 < q\phi < q\pi/2$, are at a distance of $R(\pi/2 - \phi)$ from the north pole, where R is the radius of the reference globe. Since the map preserves distances to the pole, the image of this parallel on the map will be a circle with radius equal to $R(\pi/2 - \phi)$ centered at the pole. In particular, because the length of one degree of latitude is the same on the entire globe, the images of the parallels will be evenly spaced concentric circles on the map. The base grid for this map can be constructed using only a compass and straight-edge. Fig. 8 shows the northern hemisphere.

Mathematical Foundations of GIS, Figure 8 Azimuthal equidistant projection of the northern hemisphere

Equal-Area Maps

For the display of statistical information or other data-oriented applications, it is preferable to use a base map that shows the areas of all regions of the earth's surface in their correct proportions. Such a map is called an *equal-area* or *equivalent* map by cartographers.

The total surface area of a spherical globe of radius R is $4\pi R^2$. More generally, the area of the portion of the sphere that lies between the equator and the parallel at ϕ radians is equal to $\left|2\pi R^2 \sin(\phi)\right|$. To see this, suppose that $\phi > 0$ and use the equation $z = \sqrt{R^2 - x^2 - y^2}$ for the northern hemisphere. The surface area element is

$$\begin{aligned} dA &= \sqrt{\left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2 + 1}\;\; dx\,dy \\ &= \frac{R}{\sqrt{R^2 - x^2 - y^2}}\, dx\,dy\,. \end{aligned}$$

Now integrate the surface area element over the region of integration consisting of a ring with inner radius $r = R\cos(\phi)$ and outer radius $r = R$. Switching to cylindrical coordinates yields

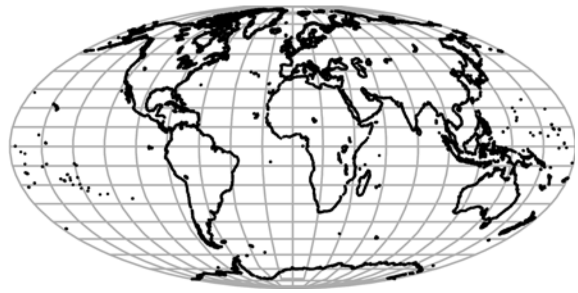
$$\begin{aligned} \text{area} &= \int_{\theta=0}^{2\pi} \int_{r=R\cos(\phi)}^{R} \frac{R \cdot r}{\sqrt{R^2 - r^2}}\, dr\, d\theta \\ &= 2\pi R^2 \sin(\phi)\,. \end{aligned}$$

Consequently, the area of the strip bounded by two parallels, with latitudes $\phi_1 < q\phi_2$, is equal to $2\pi R^2[\sin(\phi_2) - \sin(\phi_1)]$. The portion of this strip that lies between the meridians with longitudes $\theta_1 < q\theta_2$ has area

$$A_{\text{block}} = R^2 \,(\theta_2 - \theta_1)\,[\sin(\phi_2) - \sin(\phi_1)]\,, \qquad (11)$$

Since every region on the surface of the globe can be filled up, in the limit anyway, by some collection of (possibly infinitesimally small) blocks, each bounded by two parallels and two meridians, it follows that a given map projection is equal-area provided that the area of every such

Mathematical Foundations of GIS, Figure 9 Sinusoidal projection of the world



Mathematical Foundations of GIS, Figure 10 The Mollweide equal-area projection

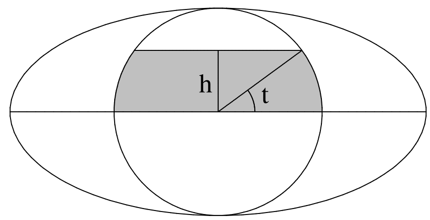
block is shown in its correct proportion. In fact, adding up the areas of many small blocks that fill up a larger region is exactly what an integral represents in calculus. One often-used equal-area projection is the sinusoidal projection, which dates back at least to 1570 when it appeared in the work of Jean Cossin of Dieppe. It was used in some later editions of Mercator's atlases (1606 to 1609) and also by Nicolas Sanson d'Abbeville beginning in 1650 and by John Flamsteed (1646–1719), the first astronomer royal of England. In addition to its current appellation, the sinusoidal projection has been known variously as the Sanson-Flamsteed, the Mercator-Sanson, or the Mercator equal-area projection.

On the sinusoidal map, the meridian at longitude θ is depicted as the graph of the curve $x = \theta \cos(y)$, for $-\pi/2 < qy < q\pi/2$, while the parallel at latitude ϕ is represented as a segment of the horizontal line $y = \phi$ extending between the map's boundary curves $x = -\pi \cos(y)$ and $x = \pi \cos(y)$. Thus, the lengths of the parallels are portrayed in their correct proportions. Also, the parallels are evenly spaced along the y-axis, just as they are along a meridian on the globe, and the meridians are spaced evenly along the parallels. For any block on the globe described by the conditions $\phi_1 < q\phi < q\phi_2$ and $\theta_1 < q\theta < q\theta_2$, the image of this block on the map has area

$$\int_{y=\phi_1}^{\phi_2} (\theta_2 - \theta_1) \cos(y)\, dy = (\theta_2 - \theta_1) [\sin(\phi_2) - \sin(\phi_1)] ,$$

in agreement with Eq. (11).

Another popular equal-area map is the Mollweide map, presented in 1805 by Karl Brandan Mollweide. This projection portrays the whole world in an ellipse whose axes are in a 2:1 ratio. The facing hemisphere is depicted as a central circle, with the diameter of the circle equal to the vertical axis of the overall ellipse. The "dark side" of the earth is split in two with one piece shown on either side of the central circle. To complete the overall structure of the map, the parallels are drawn as horizontal lines on the map, while the two meridians at θ and $-\theta$ will together form an



Mathematical Foundations of GIS, Figure 11 For each latitude ϕ, choose t so that the shaded area, $2t + \sin(2t)$, is equal to $\pi \sin(\phi)$

ellipse whose vertical axis coincides with the vertical axis of the overall ellipse. The meridians will be equally spaced along the equator.

The mathematics involved in the construction of this projection is rather more complicated than that of the sinusoidal. The central circle has radius $\sqrt{2}$ and area 2π, the area of a hemisphere on a reference globe of radius 1. If the parallel at latitude ϕ is placed on the line $y = h(\phi)$, then, to ensure that areas are preserved, the shaded region in Fig. 11 must have area $\pi \sin(\phi)$. Thus, the angle t, shown in the figure, must satisfy $\pi \sin(\phi) = 2t + \sin(2t)$. This equation can be solved only numerically. Then $h(\phi) = \sqrt{2} \sin(t)$. The meridians at $\pm\theta$ form a lune with area 4θ on a globe of radius 1 unit. The ellipse formed by these meridians on the map has a vertical axis of length $2\sqrt{2}$, and, hence, has the equation $\pi^2 x^2 + 4\theta^2 y^2 = 8\theta^2$.

In 1925, J. P. Goode introduced his Homolosine map, an interrupted projection devised by fusing together parts of four sinusoidal maps and seven Mollweide maps, with various central meridians. The Mollweide maps are used in the upper latitudes to smooth out the polar regions.

Lambert's azimuthal equal-area projection, presented by Johann Heinrich Lambert in 1772, is widely used for atlas maps today. For a map centered on the north pole, the image of the pole will be taken as the origin in the plane of the map. The parallels will be shown as concentric circles centered at the origin, with the circle corresponding to latitude ϕ having a radius of $r(\phi)$. The function $r(\phi)$, which will be decreasing, is to be determined. The meridian at

Mathematical Foundations of GIS, Figure 12 Areas are affected by scale changes

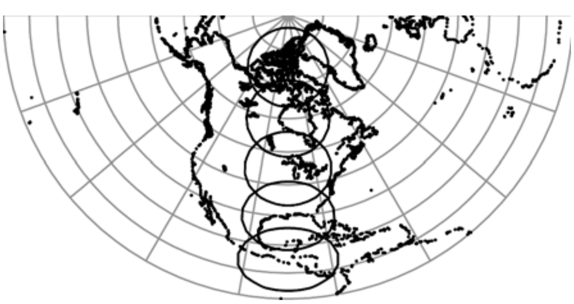
longitude θ will be portrayed as a radial line segment emanating from the origin and making an angle of θ with the positive x-axis.

While the parallel at ϕ has circumference $2\pi\cos(\phi)$ on a reference globe of radius 1, its image has circumference $2\pi r(\phi)$. Thus, the scale factor of the map along this parallel is $M_p(\phi) = r(\phi)\sec(\phi)$. The arc of any meridian lying between latitudes ϕ and $(\phi+t)$ has length t while its image on the map has length $r(\phi) - r(\phi+t)$, the gap between the circles corresponding to the two parallels. Thus, the ratio between the map measurement and the globe measurement is $(r(\phi) - r(\phi+t))/t$. Let t approach 0 to obtain the scale factor along any meridian at a point at latitude ϕ:

$$M_m(\phi) = \lim_{t \to 0} \frac{r(\phi) - r(\phi+t)}{t} = -r'(\phi)\,.$$

For this projection to preserve areas, the condition $M_p \cdot M_m = 1$ must be met. (See Fig. 12). Hence, the function $r(\phi)$ must satisfy $-\sec(\phi)r(\phi)r'(\phi) = 1$. This equation can be rewritten as $r\,\mathrm{d}r = -\cos(\phi)\,\mathrm{d}\phi$. Integrating both sides yields the equation $r^2/2 = -\sin(\phi)+C$. Since $r = 0$ when $\phi = \pi/2$, it follows that $C = 1$. Moreover, $r \geq 0$, so that $r = \sqrt{2} \cdot \sqrt{1-\sin(\phi)} = 2\sin(\pi/4 - \phi/2)$. This formula is uniquely determined by the conditions that led to it. Hence, this projection, of which an example is shown in Fig. 13, is the *only* azimuthal equal-area projection up to overall scale change.

The *conic projections* form an important class of projections that is not discussed here in detail. In these, a cone is placed on the globe and the globe somehow projected onto the cone, which is then slit open and laid out to form a sector of a circle. Conic projections were used by Ptolemy *circa* 150 A.D. and are especially useful for mapping portions of the globe that are wide east-to-west but short north-to-south, such as Russia or The United States. The United States Geological Survey uses conic projections extensively for its topographical maps. Albers' equal-area conic projection, presented in 1805, and Lambert's conformal conic, introduced in 1772, are both prevalent today. Both cylindrical and conic projections can be modified to have two standard lines where the sphere and the cylinder or cone meet. The analysis of scale factors for a conic projection is similar to that for an azimuthal projection, with the slight complication that the image fills only a sector of a circle. The angle of the sector depends on the choice of the standard lines for the map.



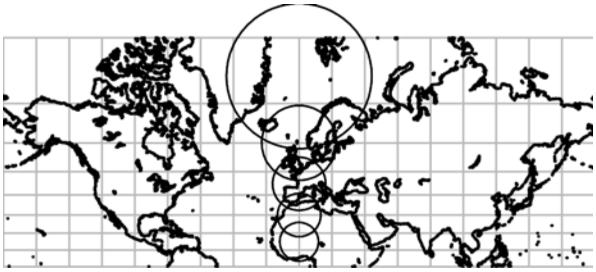
Mathematical Foundations of GIS, Figure 13 Tissot's indicatrix for Lambert's equal-area azimuthal projection. All ellipses have the same area, but are more elongated away from the pole

Map Distortion

Just as the analysis of scale factors was the key to constructing maps that preserved angles or areas, so it is central to understanding the extent to which a given map distorts those measurements. For instance, for a cylindrical, azimuthal, or conic projection, the condition $M_p = M_m$ ensures conformality, while the equation $M_p \cdot M_m = 1$ characterizes an equal-area map. More generally, for these classes of projections, the values of M_p/M_m and $M_p \cdot M_m$ can be used as measures of the distortions in angles and areas, respectively. The more these differ from the value 1, the greater are the map's distortions. For example, Mercator's map satisfies $M_p/M_m = 1$ at every point, reflecting the map's conformality. But $M_p \cdot M_m = \sec^2(\phi)$ for this map, indicating a severe distortion of ares as ϕ approaches $\pm\pi/2$. For Lambert's azimuthal equal-area projection, $M_p \cdot M_m = 1$, but $M_p/M_m = \sec^2(\pi/4 - \phi/2)$. So this map distorts angles increasingly away from the north pole.

Tissot's Indicatrix

In the late nineteenth century, a French mathematician, Tissot, developed a method, called *Tissot's indicatrix*, that has become a standard cartographic tool for quantifying and, especially, visualizing distortions in angles and areas. The starting point for this technique is Tissot's observation that, for any map projection, there is at each point on the sphere a pair of perpendicular directions whose images in the projection are also perpendicular. Tissot called these



Mathematical Foundations of GIS, Figure 14 Tissot's indicatrix for Mercator's projection. All ellipses are circles, but the areas increase away from the equator

the *principal directions* at the given point. Schematically, at each point on the map, Tissot constructed an ellipse whose principal axes were aligned with the principal directions and had lengths equal to the scale factors of the map projection in those directions. In practice, a representative selection of points is used and the ellipses are rescaled by a common factor so that they fit on the map and don't interfere with each other too much. In this way, one can make effective visual comparisons between different ellipses in the indicatrix.

For cylindrical, azimuthal, and conic projections with standard perspective, the principal directions at any point lie along the meridian and the parallel. The corresponding scale factors are simply $M_m(\phi)$ and $M_p(\phi)$. Thus, Tissot's indicatrix consists of a system of ellipses whose principal axes have lengths M_m and M_p. The area of such an ellipse is proportional to $M_p \cdot M_m$. Hence, if $M_p \cdot M_m$ is constant, then the ellipses have the same area over the whole map and the projection is equal-area. In general, the more $M_p \cdot M_m$ varies, the more the map distorts areas. Similarly, the ratio M_p/M_m represents the ratio of the lengths of the principal axes of the ellipses in the indicatrix. If this ratio is equal to 1 at every point, then every ellipse is actually a circle and the projection is conformal. The more this ratio varies from the value 1, the more elongated is the corresponding ellipse and, thus, the more distorted are the angles at that point. Tissot also used the principal scale factors to measure, at each point on the projection, the maximum error between any angle and its image angle.

Oblique Perspectives

Most maps, in their standard presentations, have either an equatorial or a polar perspective. In applications, however, a different perspective may be preferable. Conceptually, an arbitrary point A can be viewed as taking the place of the north pole and an imaginary set of "meridians" – great circles emanating from A and passing through the point antipodal to A – can be formed. Likewise, an imaginary "equator" lies halfway between A and its antipode, and "parallels" subdivide the imaginary meridians. In this new framework, every point on the globe can be assigned a new pair of relative longitude and latitude coordinates. Mathematically, the usual map equations are applied to these relative longitude and latitude values. The relative coordinates themselves can be computed using linear algebra.

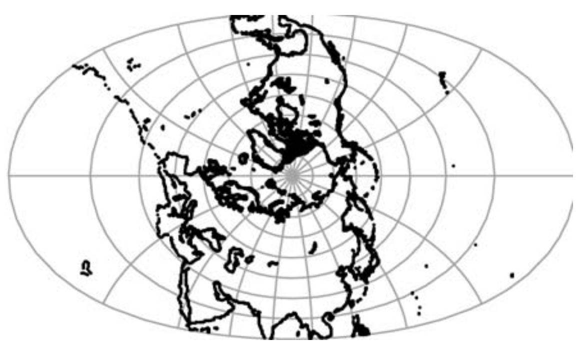
The Transverse Mercator Map

One of the most important maps that Lambert presented in 1772 is the transverse Mercator projection, a version of Mercator's map that is centered at the north pole instead of at the equator. To construct a transverse Mercator projection of the northern latitudes, the north pole, with Cartesian coordinates $N\langle 0, 0, 1\rangle$, will take the place of the point $\langle 1, 0, 0\rangle$ as the center of the map. The point $B\langle -1, 0, 0\rangle$, where the meridian at $\pm\pi$ meets the equator, will rotate into the place originally occupied by the north pole, $\langle 0, 0, 1\rangle$. Finally, the point $C\langle 0, 1, 0\rangle$ will be fixed by this rotation. The vectors N, B, and C form an orthonormal system. The matrix that converts standard Cartesian coordinates into coordinates relative to the new system is

$$T = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}.$$

The point at (θ, ϕ) has Cartesian coordinates $\langle \cos(\theta)\cos(\phi), \sin(\theta)\cos(\phi), \sin(\phi)\rangle$. Apply T to this to get the relative Cartesian coordinates $\langle \sin(\phi), \sin(\theta)\cos(\phi), -\cos(\theta)\cos(\phi)\rangle$.

For the standard Mercator map, the x-coordinate is equal to the longitude of the point being mapped. For the transverse Mercator, use the relative longitude instead: $\tilde{\theta} = \arctan\left(\frac{\sin(\theta)\cos(\phi)}{\sin(\phi)}\right)$. The y-coordinate on the map is $\ln\left|\sec(\tilde{\phi}) + \tan(\tilde{\phi})\right|$, where $\tilde{\phi} = \arcsin(-\cos(\theta)\cos(\phi))$



Mathematical Foundations of GIS, Figure 15 Transverse Mercator projection of the northern hemisphere, latitude $\phi \geq \pi/18$

is the relative latitude. Figure 15 shows a transverse Mercator projection of the northern hemisphere.

The transverse Mercator map is still conformal, though loxodromes are no longer seen as straight lines. (A path that makes a constant angle with the *relative* meridians emanating from the point B, however, is a straight line!)

In general, a map having an oblique, i. e., non-standard, perspective can be obtained by imagining the desired centering point A either as a pole or as the point where the equator and prime meridian meet. Let B be the point obtained either by adding or by subtracting $\pi/2$ from the latitude of A. To complete the orthonormal system, use either $C = A \times B$ or $C = -A \times B$, where '×' denotes the vector cross product. (The exact choices of B and C depend on the hemisphere involved and on whether the projection is equatorial or polar in its standard perspective.) Now form the transformation matrix T whose rows are given by the Cartesian coordinates of A, B, and C arranged according to the order in which they take the places of $\langle 1, 0, 0\rangle$, $\langle 0, 1, 0\rangle$, and $\langle 0, 0, 1\rangle$, respectively. The matrix T will transform standard Cartesian coordinates into relative Cartesian coordinates. The standard map equations can then be applied to the corresponding relative values of longitude and latitude.

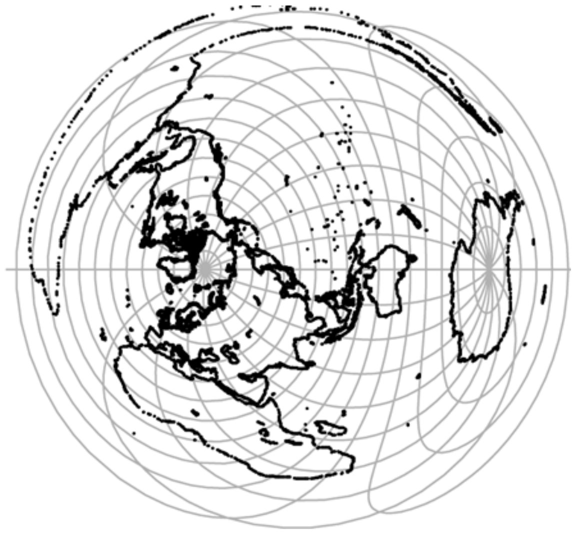
For example, using A = Tokyo, with Cartesian coordinates $\langle -.62, .52, .59\rangle$, in place of the north pole, let $B = \langle -.45, .38, -.81\rangle$ and $C = A \times B = \langle -.64, -.77, 0\rangle$. The transformation matrix is

$$T = \begin{bmatrix} -.45 & .38 & -.81 \\ -.64 & -.77 & 0 \\ -.62 & .52 & .59 \end{bmatrix}.$$

For the point with longitude θ and latitude ϕ, the relative Cartesian coordinates are

$$T \begin{bmatrix} \cos(\theta)\cos(\phi) \\ \sin(\theta)\cos(\phi) \\ \sin(\phi) \end{bmatrix} = \begin{bmatrix} -.45\cos(\theta)\cos(\phi) \\ +.38\sin(\theta)\cos(\phi) - .81\sin(\phi) \\ -.64\cos(\theta)\cos(\phi) - .77\sin(\theta)\cos(\phi) \\ -.62\cos(\theta)\cos(\phi) \\ +.52\sin(\theta)\cos(\phi) + .59\sin(\phi) \end{bmatrix}.$$

From these, the relative longitude and latitude of any point can be determined. Applying the standard formulas for an azimuthal equidistant projection to the relative coordinates yields the map in Fig. 16.



Mathematical Foundations of GIS, Figure 16 Azimuthal equidistant projection centered at Tokyo

Future Directions

Technological advances continue to yield improvements in the accuracy of GPS and other geodetic data. Though literally hundreds of map projections have been developed over the centuries, as long as people continue to find new information to present using maps there will be new ideas for projections on which to present it.

Cross References

- ► Generalization and Symbolization
- ► Map Generalization
- ► Privacy Preservation of GPS Traces
- ► Scale, Effects
- ► University of Minnesota (UMN) MapServer
- ► Wayfinding, Landmarks
- ► Web Mapping and Web Cartography

Recommended Reading

1. Banerjee, S.: Revisiting Spherical Trigonometry with Orthoginal Projectors, College Math. J. **35**(5), 375–381 (2004)
2. Berggren, J.L., Jones A.: Ptolemy's Geography. Princeton University Press, Princeton (2000)
3. Bosowski, E.F., Feeman T.G.: The use of scale factors in map analysis: an elementary approach. Cartographica **34**(4), 35–44 (1997)
4. Bugayevskiy, L.M., Snyder J.P.: Map Projections: A Reference Manual. Taylor and Francis, London (1995)
5. Cotter, C.H.: The Astronomical and Mathematical Foundations of Geography. American Elsevier, New York (1966)
6. Dent, B.D.: Cartography, 4th edn. Wm. C. Brown, Dubuque (1996)
7. Espenshade, E.B. Jr., et al. (eds.): Goode's World Atlas. 19th edn. Rand McNally, Skokie, Illinois, USA (1995)
8. Feeman, T.G.: Portraits of the Earth: A Mathematician Looks at Maps. American Mathematical Society, Providence (2002)
9. Forman S, Steen L (2006) Global Positioning System. http://www.stolaf.edu/other/ate/gps.html. Accessed 8 May 2006

10. Nord, G., Jabon, D., Nord J.: The Global Positioning System and the Implicit Function Theorem. SIAM Rev. **40**(3), 692–696 (1998)
11. Snyder, J.P.: Flattening the Earth: Two Thousand Years of Map Projections. University of Chicago Press, Chicago (1993)
12. Thompson, R.B.: Global Positioning System: The Mathematics of GPS Receivers. Math. Magazine **71**(4), 260–269 (1998)
13. Wikipedia (2006) Global Positioning System. http://en.wikipedia.org. Accessed 8 May 2006
14. Wikipedia (2006) Trilateration. http://en.wikipedia.org. Accessed 8 May 2006

Mathematical Programming

► Multicriteria Decision Making, Spatial

Mathematical Theory of Geosensor Networks

► Geosensor Networks, Formal Foundations

Matrices, Geographic

► Temporal GIS and Applications

Matrix, Inverse

► Hurricane Wind Fields, Multivariate Modeling

MAUP

► Error Propagation in Spatial Prediction

Maximum Update Interval

► Maximum Update Interval in Moving Objects Databases

Maximum Update Interval in Moving Objects Databases

Christian S. Jensen[1], Dan Lin[2], Beng Chin Ooi[2]
[1] Department of Computer Science, Aalborg University, Aalborg, Denmark
[2] Department of Computer Science, Purdue University, West Lafayette, IN, USA

Synonyms

Maximum update interval

Definition

The maximum update interval in moving objects databases denotes the maximum time duration in-between two subsequent updates of the position of any moving object. In some applications, a variation of the maximum update interval denotes the time duration within which a high percentage of objects have been updated.

Main Text

In moving objects databases, there exist a population of moving objects, where each object is usually assumed to be capable of transmitting its current location to a central server. A moving object transmits a new location to the server when the deviation between its real location and its server-side location exceeds a threshold, dictated by the services to be supported. In general, the deviation between the real location and the location predicted by the server tends to increase as time passes. Even the deviation does not increase, it is also necessary to inform the server periodically that the object still exists in the system. In keeping with this, a *maximum update interval* is defined as a problem parameter that denotes the maximum time duration in-between two updates of the position of any moving object. This definition is very helpful to index and application development especially those with functions of future prediction. Given such a time interval, trajectories of objects beyond their maximum update interval when the objects will definitely be updated usually do not need to be considered. In other words, the maximum update interval gives an idea of a time period of validity of current object information.

Cross References

► Indexing of Moving Objects, B^x-Tree
► Indexing the Positions of Continuously Moving Objects
► Indexing, BDual Tree

MB-Index

► Indexing Schemes for Multi-dimensional Moving Objects

MBR

► Minimum Bounding Rectangle

MCMC

► Hurricane Wind Fields, Multivariate Modeling

MDA

► Modeling with Enriched Model Driven Architecture

MDE

► Modeling with Enriched Model Driven Architecture

Meaning, Multiple

► Uncertainty, Semantic

Medical Geography

► Spatial Uncertainty in Medical Geography: A Geostatistical Perspective

Memory, External

► Indexing Schemes for Multi-dimensional Moving Objects

Mereotopology

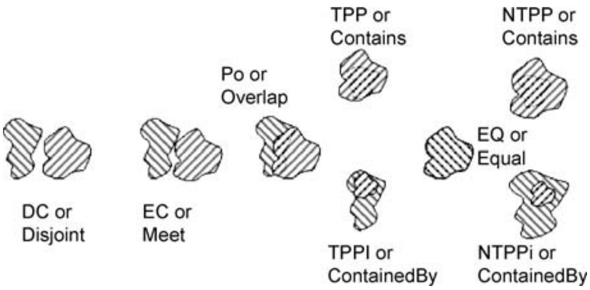
ANTHONY G. COHN
School of Computing, University of Leeds, Leeds, UK

Synonyms

Region connection calculus; RCC; 9-Intersection calculus; 4-Intersection calculus; Pointless topology

Definition

Topology, which is founded on the notion of connectedness, is at the heart of many systems of qualitative spatial relations; since it is possible to define a notion of parthood from connection, and theories of parthood are called mereologies, such combined theories are generally called mereotopologies. The best known set of relations based on a primitive notion of connectedness is the Region Connection Calculus (RCC), which defines several sets of *jointly exhaustive and pairwise disjoint, (JEPD)* relations, RCC-5, a purely mereological set, and the more widely used RCC-8 set of eight relations illustrated in Fig. 1. The primitive relation used in RCC (and several related theories) is $\mathbf{C}(x,y)$ – true when region x is connected to region y. A largely equivalent set of relations can be defined in the 4-intersection model in which relations between regions are defined in terms of whether the intersections of their



Mereotopology, Figure 1 A 2D depiction of RCC-8 relations or the eight topological relations of the 4 and 9-intersection calculi. The *arrows* show the conceptual neighborhood structure

boundaries and interiors are empty or non empty; after taking into account the physical reality of 2D space and some specific assumptions about the nature of regions, it turns out that the there are exactly eight remaining relations, which correspond to the RCC-8 relations. A generalization (the 9-intersection model) also considers the exterior of regions too, and allows further distinctions and larger sets of JEPD relations to be defined. For example, one may derive a calculus for representing and reasoning about regions in $\mathbb{Z}^2$ rather than $\mathbb{R}^2$, or between spatial entities of different dimensions (such as relations between lines and regions).

Cross References

► Conceptual Neighborhood
► Knowledge Representation, Spatial
► Representing Regions with Indeterminate Boundaries

Recommended Reading

1. Cohn, A.G., Hazarika, S.M.: Qualitative Spatial Representation and Reasoning: An Overview. Fundam. Inf. **46**(1–2), 1–29 (2001)
2. Cohn, A.G., Renz, J.: Qualitative Spatial Representation and Reasoning. In: Lifschitz, V., van Harmelen, F., Porter, F. (eds.) Handbook of Knowledge Representation, Ch. 13. Elsevier, München (2007)

Merge Designs

► Contraflow for Evacuation Traffic Management

Metadata

CHRISTOPHER J. SEMERJIAN
Institute for Environmental & Spatial Analysis, Gainesville State College, Gainesville, GA, USA

Synonyms

Geospatial metadata; Geographic metadata

Definition

A metadata record is a file of information, which captures the basic characteristics of a data or information resource. It represents the who, what, when, where, why, and how of the resource. Metadata is known as "data about data" [1]. It describes the content, quality, conditions, location, author, and other characteristics of data. Geospatial metadata is metadata that describes data or objects with geographic attributes, such as a geographic extent or a fixed location on the Earth. This geographic attribute may be a location such as latitude and longitude, a street address, or a geographic position relative to other objects. Geospatial metadata are used to document geographic digital resources such as Geographic Information System (GIS) files, raster and vector alike, and other geospatial databases [2]. Metadata makes spatial information more useful to all types of users by making it easier to document and locate data sets. Metadata helps people who use geospatial data to find the data they need and determine how best to use it [3]. It is also important because it protects the investment in data, it helps the user understand data, and it enables discovery [4]. The creation and management of metadata is both an essential and required part of GIS functionality.

Historical Background

In Greek epistemology the prefix *meta* means *about*. Thus the term metadata can be literally interpreted as "about data". Metadata is any information that describes data. Historically metadata was seen as additional information to supplement data, but not necessarily an essential part of the data. Recent advances in information technology and the rapid emergence of the digital library have somewhat altered the perception of metadata among information managers; metadata is no longer auxiliary definitions or descriptions of some library resource, but a fundamental dimension of said resource [5]. An early use of metadata in the digital world occurred in the 1960's, with the advent of the international Machine-Readable Cataloging (MARC) standards and the Library of Congress Subject Headings (LCSH) [5].

A major factor in the functionality of a Geographic Information System is data interoperability [6]. Geospatial data comes from a variety of sources in a variety of formats. Metadata is the key to maintaining interoperability by identifying standards and recording the information necessary to ensure information exchange. In the late 1970's, many government agencies in United States (US) started initiating digital mapping programs. In 1983, the Office of Management and Budget established a committee to coordinate digital cartographic activities among the US federal agencies in an effort to keep track of the enormous growth of digital geospatial data. They were to oversee any problems associated with the duplication of effort, lack of standards and inadequate interagency coordination, etc. The Office of Science and Technology Policy study recommended a centralized data base and schema to overcome such problems [1]. Thus, in 1983, the Federal Interagency Coordinating Committee on Digital Cartography (FICCDC) was established to coordinate the GIS data development activities [1]. In 1990, FICCDC was evolved into Federal Geographic Data Committee (FGDC) [7].

Beginning in 1994, Executive Order 12906 requires federal agencies to produce standardized metadata for all new geospatial data they create. The National Spatial Data Infrastructure (NSDI) was created in the same year to coordinate in collection, sharing, and use of GIS data among federal or non-government agencies [1].

Scientific Fundamentals

The FGDC promoted the development, use, sharing, and dissemination of geospatial data on a national basis [7]. The FGDC developed draft content standards for geospatial metadata in the fall of 1992. "The objectives of the standard are to provide a common set of terminology and definitions for the documentation of digital geospatial data. The standard establishes the names of data elements and compound elements (groups of data elements) to be used for these purposes, the definitions of these compound elements and data elements, and information about the values that are to be provided for the data elements" [8]. After a public comment period and revision, the FGDC approved the standard on June 8, 1994.

The standard utilizes ten categories to describe a geospatial data set. These categories are:

1. *Identification* – basic information including the title, author, abstract, purpose, geographic extent, data collection dates, status, completion date, publication date, and access constraints.
2. *Data Quality* – data quality information including positional accuracy, attribute accuracy, processing steps, and data lineage.
3. *Spatial Data Organization* – information on the spatial reference method used to depict geographic features (raster or vector) and a count of spatial features.
4. *Spatial Reference* – Horizontal and vertical coordinate system information including projection parameters, coordinate system parameters, and geodetic datum.
5. *Entity and Attribute Information* – a description of each attribute in the data table including the attribute name, data type, syntax, width, precision and domain.

6. *Distribution* – data access information including distribution formats, distribution locations, hyperlinks, and costs.
7. *Metadata Reference* – metadata compilation information including date and author.
8. *Citation Information* – the reference for the dataset including publication information and online linkages.
9. *Time Period Information* – metadata about any temporal attributes of the data set.
10. *Contact Information* – identity of contact persons or organizations associated with the data set.

Geospatial metadata will typically be stored in an XML, TXT, or HTML file with the same filename as the associated dataset. The FGDC does not specify how their ten standard metadata categories should be formatted within the metadata file. However, the FDGC does use several stylesheets for geospatial metadata. Several metadata creation tools and metadata stylesheets are available from the FGDC website or from within various geospatial software packages such as ArcGIS™, AutoDesk™, ERDAS™, and Intergraph™.

The Federal Geographic Data Committee website (http://www.fgdc.gov) is an excellent resource for geospatial metadata standards. The site contains geospatial metadata information including an overview, metadata history, importance of metadata, technical documentation, examples, and standards. A detailed description of the FGDC metadata standard can be found on the site under *Content Standard for Digital Geospatial Metadata*. A glossary and list of metadata elements are also provided.

Other organizations that have developed standards for geospatial metadata include the International Organization for Standardization (ISO) and the Open Geopstial Consortium (OGC). The International Organization for Standardization is an international standard-setting body that produces world-wide standards for commerce and industry. ISO/TC 211 is a standard technical committee formed within ISO, tasked with covering the areas of digital geographic information (such as used by geographic information systems) and geomantic [9]. Publication ISO 19115:2003 defines the schema required for describing geographic information and services; it provides information about the identification, the extent, the quality, the spatial and temporal schema, spatial reference, and distribution of digital geographic data [9]. Publication ISO 19115:2003 can be accessed through the ISO website: (http://www.isotc211.org). The Open Geospatial Consortium, Inc is an international industry consortium of 339 companies, government agencies and universities participating in a consensus process to develop publicly available interface specifications [10]. The consortium addresses issues and sets standards related to how metadata must be specified in a GIS. These recommendations and standards can be found in the Metadata WG specifications, located on the OGC website (http://www.opengeospatial.org). The geospatial metadata standards developed by the International Organization for Standardization and the Open Geospatial Consortium are tied closely to the standards developed by the Federal Geographic Data Committee and are identical in many regards. The ISO 19115 specifications are quickly becoming the world standard for geospatial metadata.

Other sources for metadata standards include the Digital Geographic Information Working Group (DGIWG). The DGWG was established in 1983 to develop standards to support the exchange of digital geographic information among nations, data producers, and data users [11]. The DGIWG published the Digital Geographic Information Exchange Standard (DIGEST) in 1994. The DIGEST can be accessed from the DGIWG website (https://www.dgiwg.org/digest/). The Massachusetts Institute of Technology Libraries Metadata Advisory Group maintains links to many other metadata standards documents. These can be accessed through the MIT Libraries website (http://libraries.mit.edu/guides/subjects/metadata/standards.html).

From a data management perspective, metadata is important for maintaining an organization's investment in spatial data. Metadata is a summary document providing content, quality, type, creation, and spatial information about a dataset. Therefore, metadata benefits an organization in the following ways:

1. Provides an inventory of data assets.
2. Helps determine and maintain the value of data.
3. Helps users and creators to determine the reliability and currentness of data.
4. Supports decision making.
5. Documents legal issues.
6. Helps keep data accurate and helps verify accuracy to support good decision making and cost savings.
7. Helps determine budgets because it provides a clearer understanding of when or if data needs to be updated or repurchased.

ESRI's GIS internet mapping service, ArcIMS™ also hosts a metadata service. This allows companies and organizations to serve geospatial metadata through web for easier public viewing. The ESRI supported spatial database engine, ArcSDE™ is the interface to the relational database that stores geospatial metadata documents for organizations. The ArcIMS metadata service uses the ArcSDE database as repository. ArcCatalog™, Metadata Explorer, Web browsers, or Z39.50 clients can access metadata stored in a metadata service.

ArcGIS has been designed to create metadata for any data set supported/created by ArcGIS as well as any other data set identified and cataloged by the user (e. g., text, CAD files, scripts). Metadata can be created for several different datasets, such as ArcInfo coverages, ESRI shape files, CAD drawings, images, grids, TINs, ArcSDE geodatabases, personal Geodatabase, maps, workspaces, folders, layers, INFO, dBASE, and DBMS tables, projections, text files, programming scripts, etc.

Key Applications

Metadata can be created for any data or information resource and is routinely used to describe or provide functional information for all types of digital data. Metadata allows more efficient query and filter applications for any digital data source. Therefore it is commonly used by libraries, corporate networks, and internet search engines for faster data retrieval and processing. Here are some of the key non-geospatial application areas for metadata.

Traditional Databases

Metadata is used in traditional database management systems such as relational database systems to store information on the size, structure, location, modification dates, and number of tables in the database system.

Operating Systems

Operating systems such as Windows or Linux store metadata on all files and folders. This includes permissions, security, display, and time stamp information for files and folders.

Digital Documents and Images

Metadata is utilized by word processors, spreadsheets, imaging software, etc. to describe key elements of the document such as creation date, modification date, author, font, spacing, size, security, etc.. Metadata is commonly generated by web-browsers, peer-to-peer software, and multimedia indexing software.

Future Directions

Most available metadata creation tools are designed to produce metadata for one data element at a time. The need exists to develop tools to manage metadata for numerous objects more effectively and efficiently [12]. Furthermore, the use of metadata is becoming more widespread and standardized and there is an increasing demand for automated metadata creation tools. Some of these tools have already been developed and are already freely available on the internet [12]. The United States Geologic Survey provides several tools, tips and tricks on their metadata information and software page [13]. Another tool used for the batch creation and maintenance of metadata within ArcGIS is available for download from the Idaho Geospatial Data Clearinghouse [12]. It is anticipated that many other metadata tools will become available in the future. By the printing of this book, many other metadata tools will be available online.

Cross References

► Data Warehouses and GIS
► Metadata and Interoperability, Geospatial

Recommended Reading

1. Wilson, R.:What is metadata? TNGIC Metadata Outreach Trainers and Power Point Modules. http://www.tnmetadata.org/training.html (2004)
2. Federal Geographic Data Committee: Geospatial Metadata. http://www.fgdc.gov/metadata (2006)
3. Federal Geographic Data Committee: Business Case for Metadata. http://www.fgdc.gov/metadata/metadata-business-case (2005)
4. Metadata Education Project, Wyoming Geographic Information Science Center. http://www.sdvc.uwyo.edu/metadata/education.html (2006)
5. United States Geologic Survey, Content Metadata Standards for Marine Science: A Case Study, USGS Open-File Report 2004-1002. http://pubs.usgs.gov/of/2004/1002/html/evol.html
6. Danko, D.: Senior Consultant, GIS Standards, Environmental Systems Research Institute, ISO Metadata Workshop
7. Federal Geographic Data Committee: The Federal Geographic Data Committee. http://www.fgdc.gov (2006)
8. Federal Geographic Data Committee: Content Standard for Digital Geospatial Metadata. http://www.fgdc.gov/standards/projects/FGDC-standards-projects/metadata/base-metadata/index_html (2006)
9. International Organization for Standardization. http://www.isotc211.org/ (2007)
10. Open Geospatial Consortium, Inc.: "About OGC". http://www.opengeospatial.org/ogc (2007)
11. Digital Geographic Information Exchange Standard (DIGEST), Version 1.2, https://www.dgiwg.org/digest, Digital Geographic Information Working Group, January (1994)
12. Idaho Geospatial Data Clearinghouse, Interactive Numeric & Spatial Information Data Engine. http://inside.uidaho.edu/whatsnew/whatsnew.htm (2006)
13. United States Geologic Survey, Formal metadata: information and software. http://geology.usgs.gov/tools/metadata/ (2006)
14. Federal Geographic Data Committee: Content Standard for Digital Geospatial Metadata Workbook. http://www.fgdc.gov/metadata/metadata-publications-list (2000)
15. Federal Geographic Data Committee: Geospatial Metadata Quick guide. http://www.fgdc.gov/metadata/metadata-publications-list (2005)
16. Federal Geographic Data Committee: Geospatial Metadata Factsheet. http://www.fgdc.gov/metadata/metadata-publications-list (2005)
17. Federal Geographic Data Committee and the National Metadata Cadre: Top Ten Metadata Errors. http://www.fgdc.gov/metadata/metadata-publications-list (2006)

18. Wayne, L., Federal Geographic Data Committee: Institutionalize Metadata Before It Institutionalizes You. http://www.fgdc.gov/metadata/metadata-publications-list (2005)
19. Wayne, L.: Metadata In Action: expanding the utility of geospatial metadata. http://www.fgdc.gov/metadata/metadata-publications-list(2005)
20. Geospatial One-Stop: Creating and Publishing in Support of Geospatial One-Stop and the National Spatial Data Infrastructure. http://www.fgdc.gov/metadata/metadata-publications-list (2004)
21. Kang-tsung Chang: Introduction to Geographic Information Systems, 4th edn., 101 (2008)
22. Wilson, R.: Tennessee Geographic Information Council, Introduction to Metadata (2004)
23. Metadata Reference Guide for Federal Geographic Data Committee Metadata, Metadata Advisory Group, Massachusetts Institute of Technology Libraries. http://libraries.mit.edu/guides/subjects/metadata/standards/fgdc.html (2004)

Metadata and Interoperability, Geospatial

DAVID M. DANKO
ESRI, Vienna, VA, USA

Synonyms

Marginalia; Summary information; Supplementary material; Legend; Catalog entry; Interoperability; ISO 19115; Interoperability, technical; Interoperability, XML schema; Semantic; Standards

Definition

Geospatial metadata is metadata about spatial information concerning objects or phenomena that are directly or indirectly associated with a location relative to the Earth; auxiliary information which provides a better understanding and utilization of spatial information. Metadata is a primary interoperability enabler.

ISO 19115 Geographic Information – Metadata defines metadata as "*data about data*." The Techweb Encyclopedia http://www.techweb.com/encyclopedia defines "data" as "*any form of information whether on paper or in electronic form. Data may refer to any electronic file no matter what the format: database data, text, images, audio and video. Everything read and written by the computer can be considered data except for instructions in a program that are executed (software).*"

Therefore, metadata is data/information in any form, paper or electronic, about data/information in any form, including computer/web service applications, no matter what format.

Historical Background

Interoperability has helped humans advance to a position as the dominate species in the world today. Interoperability becomes more complex and important as the world becomes more integrated and cultures become more interdependent. Two important forms of interoperability are technical and semantic interoperability. Geospatially humans have attained interoperability over the centuries using maps, charts, and in written and verbal descriptions. The need for geospatial interoperability is increasing as geographic information systems move into mainstream information technology (IT) applications and with the increased use of web services. There are many factors that are required to make interoperability happen; two major factors are standards and metadata. Standards: criteria which document agreement between a provider and a consumer; enable both technical and semantic interoperability. In the past standards for geographic information included those for languages, of course, and standards for consistency of scale, level of detail, geometric layout, symbology, and accuracy. With the exception of the aeronautical and hydrographic navigation fields these typically have been set by the national and commercial organizations producing the maps a charts. Metadata has always played an important role in cartography; for centuries it has provided users with an understanding of maps. Mention the word "metadata" and many think of something complex that applies only to information technology and computer science. However, metadata is not new; it is used every day in library card catalogs, Compact Disc (CD) jackets, user's manuals, and in many other ways. The field of cartography has a long history using metadata; it has been used for centuries in the margins of maps and charts. The title, source, scale, accuracy, producer, symbols, navigation notices, warnings, all of the information found in the borders of maps and charts is metadata. This metadata is very user oriented; just about anyone can pick up a map, understand the metadata, and use the map.

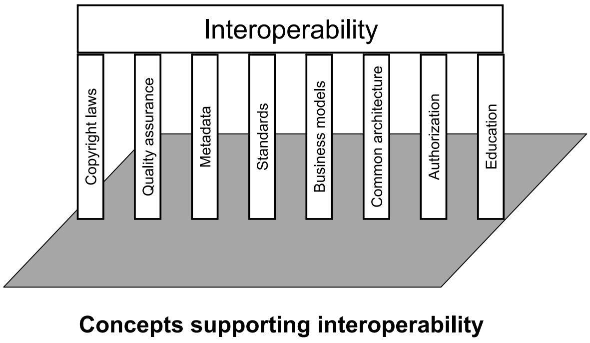
Geographic information systems (GIS) have always required interoperability. GIS uses data from multiple sources and from multiple distributed organizations within a community. For years GIS has been merging different information types: raster, vector, text, and tables. As the use of GIS grows and moves into varied disciplines the need for interoperability increases; GIS interoperates with a broad array of IT applications and is applied across diverse information communities. Web Services carry this need to new heights with loosely coupled, distributed networks.

Moving into the digital environment, metadata is equally important. Because digital data is an imperfect representation of the real world, and with the proliferation of data from an ever-widening array of sources and producers, it is important to have knowledge provided by metadata to understand, control and manage geographic information. Metadata adhering to international standards will expand interoperability enabling global networks, provide a common global understanding of geographic data, and promote global interoperability.

Moving into the world of global spatial data infrastructures, the need for internationally standardized metadata across communities was realized. In 2003 *ISO 19115 Geographic Information – Metadata* was established as an international standard it defines and standardizes a comprehensive set of metadata elements and their characteristics, along with the schema necessary to fully, and extensively, document geographic data. The standard applies to all types of geographic data and services. Since the development of the ISO metadata standard and with the wide expansion in the number of datasets and services available in the world the need for metadata focused on discovery became apparent resulting in the development of profiles of the ISO standard which support on-line catalogs, clearinghouses, and web portals.

Scientific Fundamentals

There are many things that are needed to make interoperability happen. It is necessary to have an infrastructure to support interoperability, a common architecture, and compatible technologies. Authorization (both authorization to share data and services with others, and authorization to uses other's data and services) is crucial. Ensuring that individual's and organization's intellectual property rights are not infringed is essential; therefore good copyright laws are needed. Also needed are business agreements and a business model; there must be a mutual benefit to both sides or there is no need to exchange information, no need for interoperability. Of course quality assurance helps; if the information in an exchange is not fit for purpose then there is no reason to be interoperable. **Standards** are required; standards allow us to communicate both technically – hardware and software working together; and semantically – understanding the same term for the same concept. The International Organization for Standardization Technical Committee for Geographic Information Standards (ISO/TC211) is developing an integrated suite of standards to address both technical and semantic interoperability. Of course, first and foremost is the understanding of data and services; for true interoperability, **metadata** is needed (Fig. 1). Metadata is an important part of the ISO TC 211 standards. Metadata provides a vehicle to locate and understand geospatial data which may be



Metadata and Interoperability, Geospatial, Figure 1

produced by one community and applied by another. As humans move into the age of global spatial data infrastructures, knowledge about widely distributed and dissimilar geographic data is essential to universally allow users to locate, evaluate, extract, and employ the data. Varied and wide spread communities with a common understanding of metadata will be able to manage, share, and reuse each other's geographic data, making global interoperability a reality. An international metadata standard provides this common understanding worldwide. Metadata standards provide pick-lists of metadata elements so that producers will know what metadata to collect and users will know what metadata to look for. The pick-list includes a vocabulary fully defining the metadata elements so that producers and users around the globe can understand the metadata.

The ISO 19115 Metadata standard defines and standardizes a comprehensive set of metadata elements and their characteristics, along with the schema necessary to fully, and extensively, document geographic data. The standard applies to all geographic data – it is applicable to datasets in series, datasets, individual geographic features, and their attributes. The standard defines the minimum set of metadata required to serve the wide range of metadata applications, as well as optional metadata elements to support a more extensive description of geographic data. Because of the diversity of geographic data, no single set of metadata elements will satisfy all requirements; for this reason the ISO metadata standard provides a standardized way for users to extend their metadata and still ensure interoperability allowing other users to comprehend and exploit this extended metadata.

Many geographic metadata standards have been in existence prior to the development of this ISO standard. In many cases these separate information community, regional, and national standards evolved in separate niches and are incompatible. Several general metadata standards that do provide minimal global interoperability do not adequately support geographic information. This incompatibility and insufficiency was the motivation for the development of ISO 19115. The ISO metadata standard was designed:

- to support geographic information;
- to work with wider information technology standards and practices;
- to serve the global community, in a multi-national, multi-language environment;
- based on a foundation of national, regional, and special information community standards and experiences and a thorough requirements analysis, and implementation testing.

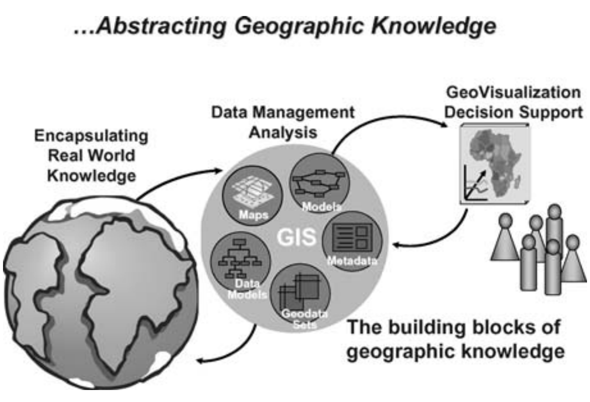
Geographic information systems encapsulate real world geographic knowledge into an information system environment by abstracting geographic knowledge into 5 basic building blocks allowing manipulation, data management, and analysis to support geo-visualization and decision making. These 5 building block element are: *data models, geodata sets, processes and workflows, maps and globes, and metadata*. *Data models* use spatial schemas, methods for defining/encapsulating geometry, topology, and networks, typically using a standardized modeling language or following rules for application schemas, to produce a template defining the relationships, rules, object definitions, and behavior of an abstraction of a universe of discourse for a specific user's conceptual view of geographic reality. *Geodata sets* are an instantiation of these models with digital data, typically in raster or vector form. Not all geographic phenomena can be abstracted using data models some are the result of a process and must be modeled using *process and workflow models*. *Maps and globes* are of course the oldest form of abstracting real world geography through the graphical display of geometry, topology, and attribution on paper, computer monitors, physical and virtual globes, and other display technology. All other aspects of real world geographic knowledge that cannot be modeled using the above 4 elements must be described using *metadata* (Fig. 2). Any description or abstraction of reality is always partial and always just one of many possible "views." This view, or model, of the real world is not an exact duplication; some things are approximated, others are simplified, and some things are ignored – there is no such thing as perfect, complete, and correct data. To insure that data is not misused, the assumptions and limitations affecting the collection of the data must be fully documented. Metadata allows a producer to fully describe their geospatial data; users can understand the assumptions and limitations and evaluate the dataset's applicability for their intended use.

Metadata serves four purposes:

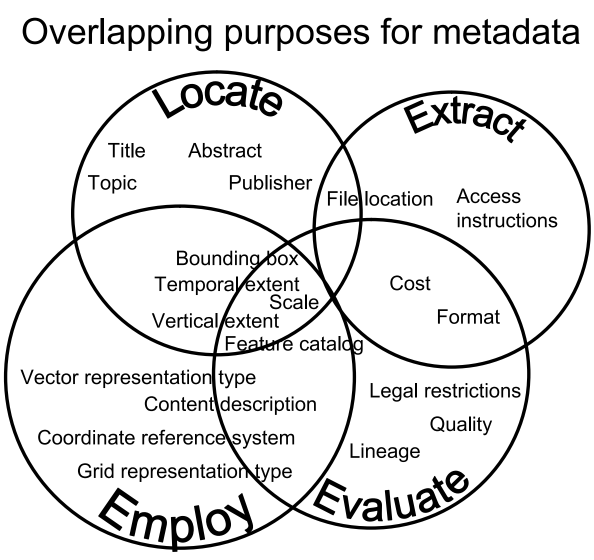
Locate: Metadata enables users to locate geospatial information and allows producers to "advertise" their data or service. Metadata helps organizations locate data outside their organization and find partners to share in data collection and maintenance.

Evaluate: By having proper metadata elements describing a dataset or service, users are able to determine if it will be suitable for their intended use. Understanding the quality and accuracy, the spatial and temporal schema, the content, and the spatial reference system used, allows users to determine if a dataset fills their needs. Metadata also provides the size, format, distribution media, price, and restrictions on use, which are also evaluation factors.

Extract: After locating a dataset and determining if it meets user's needs, metadata is used to describe how to access a dataset and transfer it to a specific site. Once it



Metadata and Interoperability, Geospatial, Figure 2



Metadata and Interoperability, Geospatial, Figure 3

has been transferred, users need to know how to process and interpret the data and incorporate it into their holdings.
Employ: Metadata is needed to support the processing and the application of a dataset. Metadata facilitates proper utilization of data, allowing users to merge and combine data with their own, apply it properly, and have a full understanding of its properties and limitations (Fig. 3).
Metadata must be collected on all products (geospatial and non-geospatial) and should be produced – when knowledge of the products is fully understood – at the time of data production.

Key Applications

Metadata is required in at least four different circumstances and perhaps in different forms to facilitate its use: in a catalog for data discovery purposes; embedded within a dataset for direct use by application software; in a historical archive; and in a human readable form to allow users to understand and get a "feel" for the data they are using.
Catalogs: Metadata for cataloging purposes should be in a form not unlike a library card catalog or on-line catalog. Metadata in a catalog should support searches by subject matter/theme, area coverage/location, author/producer, detail/resolution/scale, currency/date, data structure/form, and physical form/media.
Historical Records: Metadata should support the documentation of data holdings to facilitate storage, updates, production management, and maintenance of geospatial data. Historical records provide legal documentation to protect an organization if conflicts arise over the use or misuse of geospatial data.
Within a geospatial dataset: Metadata should accompany a dataset and be in a form to support the proper application of geospatial data. GIS and other application software using data need to evaluate data as it applies to a situation. In this form the metadata may be incorporated into the structure of the data itself.
In a human readable form: Metadata in a form in which a computer can locate, sort, and automatically process geospatial data greatly enhance its use, but eventually a human must understand the data. One person's, or organization's, geospatial data is a subjective abstract view of the real world, it must be understood by others to ensure the data is used correctly. Metadata needs to be in a form which can be readily and thoroughly understood by users.
Non-geographers using geospatial data: A revival in the awareness of the importance of geography and how things relate spatially, combined with the advancement in the use of electronic technology, have caused an expansion in the use of digital geospatial information and geographic information systems (GIS) worldwide. Increasingly, individuals from a wide range of disciplines outside of the geographic sciences and information technologies are capable of producing, enhancing, and modifying digital geospatial information. As the number, complexity, and diversity of geospatial datasets grow, the use of metadata providing an understanding of all aspects of this data grows in importance.
Increasingly, the producer is not the user: Most geospatial data is used multiple times, perhaps by more than one person. Typically, it is produced by one individual or organization and used by another. Proper documentation provides those not involved with data production with a better understanding of the data and enable them to use it properly. As geospatial data producers and users handle more and more data, proper metadata documentation provides them with a keener knowledge of their holdings and allows them to better manage data production, storage, updating, and reuse.

Future Directions

As stated above, metadata is a primary interoperability enabler by making it possible for geospatial information users to better understand their information. Although metadata in the past played a key role in the production and use of paper maps and navigation charts, with the advent of the digital age, metadata has often been overlooked. People have been more concerned with the process of encapsulating real world knowledge into an information system. Now that this process of modeling one's "universe of discourse" has become routine and easily achievable using geographic information systems and metadata standards have been produced which guide producers and users on the importance, the definition, the concepts, and the utilization of geospatial metadata, it is now increasingly being produced and utilized. With the standardization of metadata many GIS and data collection systems are developing tools that automate metadata collection and management. The scale, coordinate reference system, language, character-set, keywords, data dictionary, and other information available in the data or the information system can be automatically collated into a metadata file.
With the recent development of *ISO/TS 19139 Metadata – XML Schema* technical interoperability has been achieved, enabling the exchange and machine parsing of metadata. ISO/TS 19139 also provides the capability to furnish multi-lingual metadata and enables the use of pre-defined code-lists standardizing information fields with vocabularies tailored for specific cultures or disciplines.
Since the development of an international geospatial metadata standard nations, regions, and scientific, defense, and commercial disciplines have been establishing profiles of

this standard. Profiles allow information communities to tailor the standard to be "tuned" to meet the needs of a specific society or discipline. Profiles tailored for language, culture and specific vocabularies are being developed for Europe, North America, Latin America, and for information communities such as defense, environment, biology, navigation communities, and others. This development of profiles of ISO 19115 will continue – refining the understanding of metadata and the data it is describing - increasing interoperability across the globe and across information communities.
In the past, geospatial datasets – typically a full range of themes and/or feature types – were collected from a common source, scanning maps or data extraction from a single mono image or stereo model; metadata covering a specific dataset was fully adequate. Increasingly, now and in the future, geographic information is being gathered from a wide variety of sources and geospatial datasets and data bases are being incrementally updated necessitating the use of feature level and hierarchical levels of metadata; metadata about specific datasets as well as metadata about specific features within the datasets. As database and storage techniques improve feature level metadata will become more common.

Cross References

- ▶ Metadata
- ▶ OGC's Open Standards for Geospatial Interoperability

Recommended Reading

1. Federal Geographic Data Committee (FGDC). Geospatial Metadata. http://www.fgdc.gov/metadata
2. Moellering, H. (ed.): World Spatial Metadata Standards. Elsevier, Oxford. (2005)
3. Dangermond, J.: Speaking the Language of Geography – GIS, ArcNews. Fall (2004). http://www.esri.com/news/arcnews/fall04articles/speaking-the-language1of2.html
4. Chan, L.M., Zeng, M.L.: Metadata Interoperability and Standardization – A Study of Methodology Part 1, D-Lib Magazine. **12**(6) (2006). ISSN 1082–9873. http://www.dlib.org/dlib/june06/chan/06chan.html
5. National Aeronautics and Space Administration Geospatial Interoperability Office: Geospatial Interoperability Return on Investment Study, Greenbelt, MD, USA, April (2005). http://gio.gsfc.nasa.gov/docs/ROI%20Study.pdf
6. Longhorn, R.: Geospatial Standards, Interoperability, Metadata Semantics and Spatial Data Infrastructure, background paper for NIEeS Workshop on Activating Metadata, Cambridge, UK 6–7 July 2005. http://archive.niees.ac.uk/talks/activating_metadata/Standards_Overview_and_Semantics.pdf
7. National Information Standards Organization: Understanding Metadata. NISO Press, Bethesda, MD (2004). ISBN: 1-880124-62-9. http://www.niso.org/standards/resources/UnderstandingMetadata.pdf

Methods of Photogrammetry

▶ Photogrammetric Methods

Microgeomatics

▶ Indoor Positioning

Minimum Bounding Rectangle

Jordan Wood
Department of Computer Science,
University of Minnesota, Minneapolis, MN, USA

Synonyms

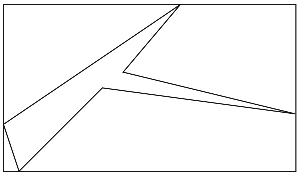
MBR; MOBR; Minimum orthogonal bounding rectangle

Definition

A minimum bounding rectangle is used to approximate a more complex shape. It is a rectangle whose sides are parallel to the *x* and *y* axises and minimally enclose the more complex shape.

Main Text

Spatial objects can take a significant amount of memory to represent. For example, a polygon which represents the borders of a country could have tens of thousands of vertices. A polyline which represents a complex linear feature such as a river would also have many vertices. Doing geometric operations such as finding objects which overlap such a complex object would be very computationally expensive, since the location of every vertex would have to be considered. There are times when we only need to know the approximate geometrical features of an object, such as during the filter step of a filter and refine strategy. In these cases, the minimum bounding rectangle (MBR) is used to approximate the shape in a simpler manner. The sides of an MBR are always parallel to the *x* and *y* axises of the space in question. Also, it is the smallest rectangle with this property which completely encloses the original shape. It can be calculated and stored as the minimum and maximum x and y values of the original shape. An example of an MBR is shown in Fig. 1. The rectangle is the MBR of the polygon.



Minimum Bounding Rectangle, Figure 1

Cross References

► Indexing Spatial Constraint Databases
► Plane Sweep Algorithm

Recommended Reading

1. Shekhar, S., Chawla, S.: Spatial Databases, A Tour, Pearson Education, Inc., New Jersey, USA (2003) ISBN 0-13-017480-7

Minimum Bounding Rectangles

► Oracle Spatial, Geometries

Minimum Orthogonal Bounding Rectangle

► Minimum Bounding Rectangle

Mining Collocation Patterns

► Co-location Patterns, Algorithms

Mining Sequential Patterns from Spatio-Temporal Databases

► Sequential Patterns, Spatio-temporal

Mining Spatial Association Patterns

► Co-location Patterns, Algorithms

Mining Spatio-Temporal Datasets

► Trajectories, Discovering Similar

MLPQ Spatial Constraint Database System

PETER REVESZ
Department of Computer Science and Engineering,
University of Nebraska-Lincoln, Lincoln, NE, USA

Synonyms

Management of linear programming queries; MLPQ system

Definition

The MLPQ system [1,2,3] is a spatial constraint database system developed at the University of Nebraska-Lincoln. The name of the system is an acronym for *Management of Linear Programming Queries*. The special feature of the MLPQ system is that it can run both SQL and Datalog queries on spatial constraint databases with linear inequality constraints. The MLPQ system is applied in the area of geographic information systems, moving objects databases and operations research. The MLPQ system has an advanced graphical user interface that displays maps and animates changes over time. Using a library of special routines, MLPQ databases can be easily made web-accessible.

Main Text

The MLPQ system accepts input textfiles specified as a set of constraint tuples or facts. Each atomic constraint must written in a way that all variables are on the left hand side of the >= comparison operator meaning "greater than or equal." For example, the Lincoln town area can be represented as follows.

```
begin %Lincoln-area%
Lincoln (id, x, y, t) :- id = 1, y − x >= 8, y >= 14, x >= 2, −y >= − 18,
                         −y − z >= −24.
Lincoln (id, x, y, t) :- id = 2, x − y >= −8, 0.5 x − y >= −12,
                         −y >= −18, x >= −14, y >= 8, 3x + y >= 32.
end %Lincoln-area%
```

The MLPQ allows browsing of the directory of input data files. Once an input datafile is selected, then the system immediately displays it on the computer screen. For queries and advanced visualization tools a number of graphical icons are available. For example, when clicking on the icon that contains the letters `SQL` a dialog box will be called. The dialog box allows selection of the type of SQL query that the user would like to enter. The types available are basic, aggregate, nested, set, and recursive SQL queries. Suppose one clicks on `AGGREGATE`. The a new dialog box will prompt for all parts of an aggregate SQL query, including the part that creates an output file with a specific name as the result of executing the query. For example, one can enter either of the SQL queries that one saw in the entry on Spatial Constraint Databases. When the SQL query is executed and the output relation is created, it is shown on the left side of the computer screen. Clicking on the name of the output relation prompts MLPQ to display it as a spatial constraint database table and/or as a map if that is possible. MLPQ requires each constraint tuple to have an id as its first field (this helps internal storage and indexing); however, any number of spatiotemporal and non-spatiotemporal fields can follow the id field. If there are more than two spatial fields, then the projections to the first two spatial fields (assumed to be the second and third fields) are displayed only. To visualize moving objects with a growing or shrinking spatial area, special animation algorithms need to be called. The system is available free from the author's webpage at cse.unl.edu/~revesz.

Cross References

► Constraint Database Queries
► Indexing Spatial Constraint Databases
► Visualization of Spatial Constraint Databases

Recommended Reading

1. Kanjamala, P., Revesz, P., Wang, Y.: MLPQ/GIS: A GIS using linear constraint databases. Proc. 9th COMAD International Conference on Management of Data, pp. 389–393. McGraw-Hill, New Delhi (1998)
2. Revesz, P., Chen, R., Kanjamala, P., Li, Y., Liu, Y., Wang, Y.: The MLPQ/GIS Constraint Database System. In: Proc. ACM SIGMOD International Conference on Management of Data. ACM Press, New York (2000)
3. Revesz, P., Li, Y.: MLPQ: A linear constraint database system with aggregate operators. In: Proc. 1st International Database Engineering and Applications Symposium, pp. 132–7. IEEE Press, Washington (1997)

MLPQ System

► MLPQ Spatial Constraint Database System

Mobile Maps

► Web Mapping and Web Cartography

Mobile Object Indexing

GEORGE KOLLIOS[1], VASSILIS TSOTRAS[2], DIMITRIOS GUNOPULOS[2]

[1] Department of Computer Science, Boston University, Boston, MA, USA

[2] Department of Computer Science and Engineering, University of California-Riverside, Riverside, CA, USA

Synonyms

Spatio-temporal index; Indexing moving objects

Definition

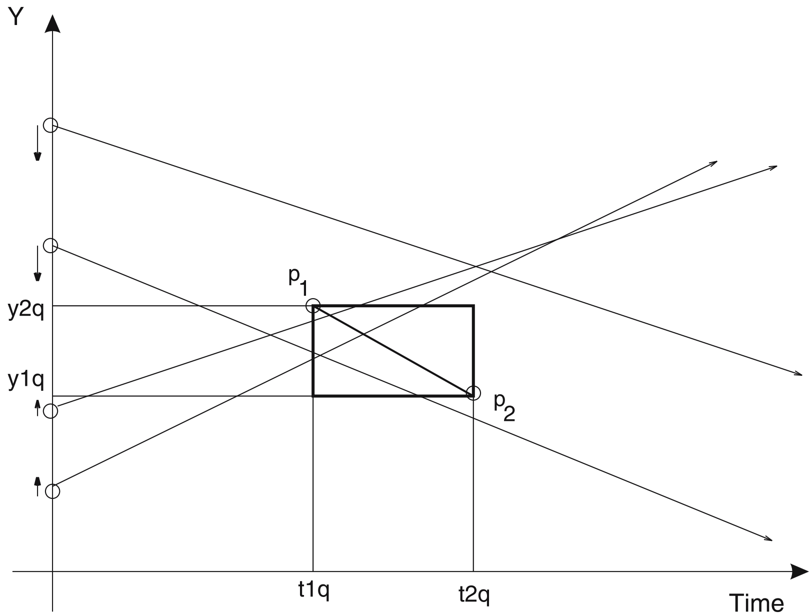
Consider a database that records the position of mobile objects in one and two dimensions, and following [8, 13,16], assume that an object's movement can be represented (or approximated) with a linear function of time. For each object, the system stores an initial location, a starting time instant and a velocity vector (speed and direction). Therefore, the future positions of the object can be calculated, provided that the characteristics of its motion remain the same. Objects update their motion information when their speed or direction changes. It is assumed that the objects can move inside a finite domain (a line segment in one dimension or a rectangle in two). Furthermore, the system is dynamic, i. e., objects may be deleted or new objects may be inserted.

Let $P(t_0) = [x_0, y_0]$ be the initial position of an object at time t_0. Then, the object starts moving and at time $t > t_0$ its position will be $P(t) = [x(t), y(t)] = [x_0 + v_x(t - t_0), y_0 + v_y(t - t_0)]$, where $V = [v_x, v_y]$ is its velocity vector. An example for the one-dimensional case is shown in Fig. 1.

Range predictive queries in this setting have the following form: "Report the objects located inside the rectangle $[x_{1q}, x_{2q}] \times [y_{1q}, y_{2q}]$ at the time instants between t_{1q} and t_{2q} (where $t_{now} \le t_{1q} \le t_{2q}$), given the current motion information of all objects" (i. e., the *two-dimensional Moving Objects Range (MOR) query* [8]).

Historical Background

The straightforward approach of representing an object moving on an one-dimensional line is by plotting the trajectories as lines in the time-location (t,y) plane (same for (t, x) plane). The equation describing each line is $y(t) = vt + a$, where v is the slope (velocity in this case) and a is the intercept, which is computed using the motion information (Fig. 1). In this setting, the query is expressed as the two-dimensional interval $[(y_{1q}, y_{2q}), (t_{1q}, t_{2q})]$ and it reports the objects that correspond to the lines intersecting the query rectangle. The space-time approach provides an intuitive representation. Nevertheless, it is problematic since the trajectories correspond to very long lines (going to infinity). Using traditional indexing techniques in this setting tends to reveal many drawbacks. A method that is based on this approach partitions the space into disjoint cells and stores those lines that intersect it in each cell [4,15]. The shortcoming of these methods is that they introduce replication since each line is copied into the cells that intersect it. Given that lines are typically long, the situation becomes even worse. Moreover, using space partitioning would also result in high update overhead since when an object changes its motion information, it has to be removed from all cells that store its trajectory.



Mobile Object Indexing, Figure 1
Trajectories and query in (t, y) plane

Saltenis et al. [13] presented another technique to index moving objects. They proposed the time-parametrized R-tree (TPR-tree), which extends the R*-tree. The coordinates of the bounding rectangles in the TPR-tree are functions of time and, intuitively, are capable of following the objects as they move. The position of a moving object is represented by its location at a particular time instant (reference position) and its velocity vector. The bounding intervals employed by the TPR-tree are not always minimum since the storage cost would be excessive. Even though it would be the ideal case (if the bounding intervals were kept always minimum), doing so could deteriorate to enumerating all the enclosed moving points or rectangles. Instead, the TPR-tree uses "conservative" bounding rectangles, which are minimum at some time point, but not at later times. The bounding rectangles may be calculated at load-time (i. e., when the objects are first inserted into the index) or when an update is issued. The TPR-tree assumes a predefined time horizon H, from which all the time instances specified in the queries are drawn. This implies that the user has good knowledge of (or can efficiently estimate) H. The horizon is defined as $H = UI + W$, where UI is the average time interval between two updates and W is the querying window. The insertion algorithm of the R*-tree, which the TPR-tree extends to moving points, aims at minimizing objective functions such as the areas of the bounding rectangles, their margins (perimeters), and the overlap among the bounding rectangles. In the case of the TPR-tree, these functions are time dependent and their evolution in $[t_l, t_l + H]$ is considered where t_l is the time instance when the index is created. Thus, given an objective function $A(t)$, instead of minimizing the objective function, the integral $\int_{t_l}^{t_l+H} A(t)\,\mathrm{d}t$ is minimized. An improved version of the TPR-tree, called TPR*-tree, was proposed by Tao et al. [14]. The authors provide a probabilistic model to estimate the number of disk accesses for answering predictive window range queries on moving objects and using this model they provide a hypothetical "optimal" structure for answering these queries. Then, they show that the TPR-tree insertion algorithm leads to structures that are much worse than the optimal one. Based on that, they propose a new insertion algorithm, which, unlike the TPR-tree, considers multiple paths and levels of the index in order to insert a new object. Thus, the TPR*-tree is closer to the optimal structure than the TPR-tree. The authors suggest that although the proposed insertion algorithm is more complex than the TPR-tree insertion algorithm, it creates better trees (MBRs with tighter parametrized extends), which leads to better update performance. In addition, the TPR*-tree employs improved deletion and node splitting algorithms that further improve the performance of the TPR-tree. The STAR-tree, introduced by Procopiuc et al. [12], is also a time-parametrized structure. It is based upon R-trees, but it does not use the notion of the horizon. Instead, it employs kinetic events to update the index when the bounding boxes start overlapping a lot. If the bounding boxes of the children of a node v overlap considerably, it re-organizes the grand children of v among the children of v. Using geometric approximation techniques developed in [1], it maintains a time-parametrized rectangle $A_v(t)$, which is a close approximation of $R_v(t)$, the actual minimum bounding rectangle of node v at any time instant t in to the future. It provides a trade-off between the quality of $A_v(t)$ and the complexity of the shape of $A_v(t)$. For linear motion, the trajectories of the vertices of $A_v(t)$ can be represented as polygonal chains. In order to guarantee that $A_v(t)$ is an ε-approximation of $R_v(t)$, trajectories of the corners of $A_v(t)$ need $O(1/\sqrt{\varepsilon})$ vertices. An ε-approximation means that the projection of the $A_v(t)$ on (x, t) or (y, t) planes contains the corresponding projections of $R_v(t)$, but it is not larger than $1+\varepsilon$ than the extend on the $R_v(t)$ at any time instant.

Finally, another approach to index moving objects is based on the dual transformation that is discussed in detail in the following text. In particular, the *Hough-X representation* of a 2D moving point o is a four-dimensional vector (x, y, v_x, v_y) where x and y is the location of the moving object at the reference time t_{ref} and v_x and v_y is the velocity of the object projected on the x and y axes. Based on this approach, Agarwal et al. [2] proposed the use of multi-level partition trees[1] to index moving objects using the duality transform in order to answer range queries at a specific time instant (i. e., snapshot queries, where $t_{1q} = t_{2q}$). They decompose the motion of the objects on the plane by taking the projections on the (t, x) and (t, y) planes. They construct a primary partition tree T^x to keep the dual points corresponding to the motion projected on the (t, x) plane. Then, at every node v of T^x, they attach a secondary partition T^y_v for the points S^y_v with respect to the (t, y) projection, where S_v is the set of points stored in the primary subtree rooted at v. The total space used by the index is $O(n \log_B n)$, where N is the number of objects, B is the page capacity and $n = \mathrm{N/B}$. The query is answered by decomposing it into two sub-queries, one on each of the two projections, and taking the dual of them, σ^x and σ^y, respectively. The search begins by searching the primary partition T^x for the dual points, with respect to the (t, x) projection, that satisfy the query σ^x. If it finds a triangle associated with a node v of the partition tree T^x that lies completely inside σ^x, then it continues searching in the secondary tree T^y_v and reports all dual points, with respect to (t, y) projection, that satisfy the query σ^y. The query is satisfied if and only if the query in both projections is satisfied. This is true for snapshot range queries. In [2], it is shown that the query takes $O(n^{\frac{1}{2}+\varepsilon} + K/B)$ I/Os (here K is the size of the query result) and that the size of the index can be reduced to $O(n)$ without affecting the asymptotic query time. Furthermore, by using multiple multilevel partition trees, is is also shown that the same bounds hold for the window range query. Elbassioni et al. [5] proposed a technique (MB-index) that partitions the objects along each dimension in the dual space and uses B+-trees in order to index each partition. Assuming a set of N objects moving in d-dimensional space with uniformly distributed and independent velocities and initial positions, they proposed a scheme for selecting the boundaries of the partitions and answering the query, yielding a $O(n^{1-1/3d} * (\sigma \log_B n)^{1/3d} + k)$ average query time using $O(n)$ space ($n = N/B$, $k = K/B$). The total number of B-trees used is $\sigma 3^d s^{2d-1}$, where $\sigma = \prod_{i=1}^{d} \ln(v_{i,max}/v_{i,min})$ and $s = (n/\log_B n)^{1/d}$, where $v_{i,\max}$ and $v_{i,\min}$ are the maximum and minimum velocities in dimension i respectively. The dual transformation has been adapted in [11], where the advantages over the TPR-trees methods have also been observed. Using the idea in [8], trajectories of d-dimensional moving objects are mapped into points in the dual 2d-dimensional space and a PR-quadtree is built to store the 2d-dimensional points. Similarly with [8], a different index is used for each of two reference times that change at periodic time intervals. At the end of each period, the old index is removed and a new index with a new reference point is built. Yiu et al. [17] proposed the B^{dual}-tree that uses the dual transformation and maps the dual points to their Hilbert value, thereafter using a B+-tree to index the objects. That was an improvement of the method proposed by Jensen et al. [7], which indexes the Hilbert value of the locations of the moving objects (without taking into account their velocities).

Scientific Fundamentals

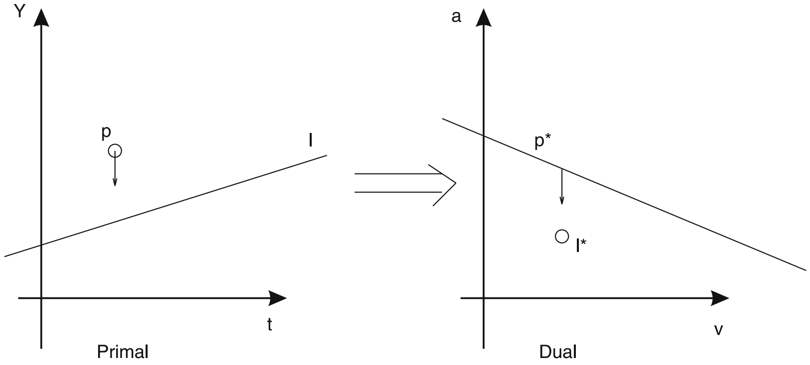
The Dual Space-Time Representation

In general, the dual transformation is a method that maps a hyper-plane h from R^d to a point in R^d and vice-versa. In this section, it is briefly described how the problem at hand can be addressed in a more intuitive way by using the dual transform for the one-dimensional case.

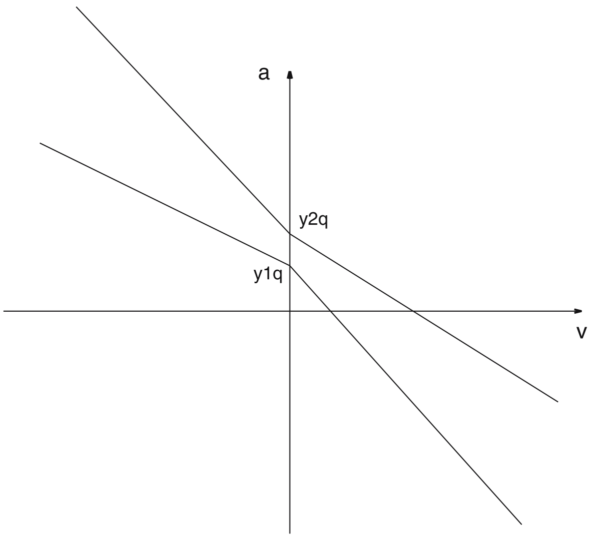
Specifically, a line from the primal plane (t, y) is mapped to a point in the dual plane. A class of transforms with similar properties may be used for the mapping. The problem setting parameters determine which one is more useful.

One dual transformation for mapping the line with equation $y(t) = vt + a$ to a point in R^2 is to consider the dual plane where one axis represents the slope of an object's trajectory (i. e., velocity) and the other axis its intercept (Fig. 2). Thus, the dual point is (v, a) (this is called Hough-X transform). Similarly, a point $p = (t, y)$ in the primal space is mapped to line $a(v) = -t\,v + y$ in the dual space. An important property of the duality transform is that it preserves the above-below relationship. As it is shown in Fig. 2, the dual line of point p is above the dual point l^* of the line l. Based on the above property, it is easy to show that the 1-d query $[(y_{1q}, y_{2q}), (t_{1q}, t_{2q})]$ becomes a polygon in the dual space. Consider a point moving with positive

[1] Partition trees group a set of points into disjoint subsets denoted by triangles. A point may lie into many triangles, but it belongs to only one subset.



Mobile Object Indexing, Figure 2 Hough-X dual transformation: primal plane (*left*), dual plane (*right*)



Mobile Object Indexing, Figure 3 Query on the Hough-X dual plane

velocity. Then, the trajectory of this point intersects the query if and only if it intersects the segment defined by the points $p_1 = (t_{1q}, y_{2q})$ and $p_2 = (t_{2q}, y_{1q})$ (Fig. 1). Thus, the dual point of the trajectory must be above the dual line p_2^* and below p_1^*. The same idea is used for the negative velocities.

Therefore, using a linear constraint query [6], the query Q in the dual Hough-X plane (Fig. 3) is expressed in the following way:

If $v > 0$, then $Q = C_1 \wedge C_2$,
where: $C_1 = a + t_{2q} v \geq y_{1q}$ and $C_2 = a + t_{1q} v < q y_{2q}$
If $v < 0$, then $Q = D_1 \wedge D_2$,
where: $D_1 = a + t_{1q} v \geq y_{1q}$ and $D_2 = a + t_{2q} v < q y_{2q}$

By rewriting the equation $y = vt + a$ as $t = \frac{1}{v} y - \frac{a}{v}$, a different dual representation can be used. Now the point in the dual plane has coordinates (b, n), where $b = -\frac{a}{v}$ and $n = \frac{1}{v}$ (Hough-Y). Coordinate b is the point where the line intersects the line $y = 0$ in the primal space. By using this transform, horizontal lines cannot be represented. Similarly, the Hough-X transform cannot represent vertical lines. Therefore, for static objects, only the Hough-X transform can be used.

Indexing in One Dimension

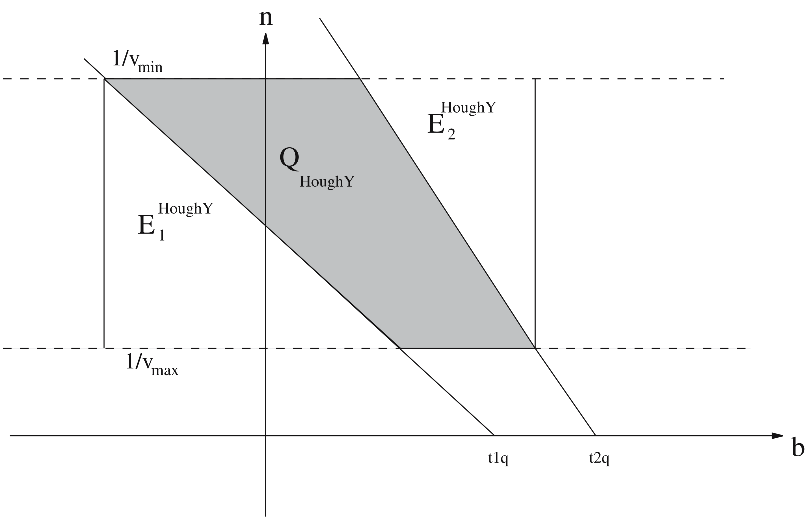
In this section, techniques for the one-dimensional case are presented, i. e., for objects moving on a line segment. There are various reasons for examining the one-dimensional case. First, the problem is simpler and can give good intuition about the various solutions. It is also easier to prove lower bounds and approach optimal solutions for this case. Moreover, it can have practical uses as well. A large highway system can be approximated as a collection of smaller line segments (this is the 1.5 dimensional problem discussed in [8]), on each of which the one-dimensional methods can be applied.

An (Almost) Optimal and Not Practical Solution

Matousek [9] gave an almost optimal algorithm for simplex range searching given a static set of points. This main memory algorithm is based on the idea of simplicial partitions.

For a set S of N points, a simplicial partition of S is a set $\{(S_1,\Delta_1), \ldots (S_r,\Delta_r)\}$ where $\{S_1, \ldots, S_r\}$ is a partitioning of S and Δ_i is a triangle that contains all the points in S_i. If $\max_i |S_i| < 2 \min_i |S_i|$, where $|S_i|$ is the cardinality of the set S_i, the partition is balanced. Matousek [9] shows that, given a set S of N points and a parameter s (where $0 < s < N/2$), it can be constructed in linear time a balanced simplicial partition for S of size $O(s)$ such that any line crosses at most $O(\sqrt{s})$ triangles in the partition.

This construction can be used recursively to construct a partition tree for S. The root of the tree contains the whole set S and a triangle that contains all the points. Then, a balanced simplicial partition of S of size $\sqrt{|S|}$ is found.



Mobile Object Indexing, Figure 4 Query on the dual Hough-Y plane

Each of the children of the root are associated with a set S_i from the simplicial partition and the triangle Δ_i that contains the points in S_i. For each of the S_i's, simplicial partitions of size $\sqrt{|S_i|}$ are computed and continue to until each leaf contains a constant number of points. The construction time is $O(N \log_2 N)$.

To answer a simplex range query, the procedure starts at the root. Each of the triangles in the simplicial partition at the root checks if (i) it is inside the query region, (ii) it is outside the query region or, (iii) it intersects one of the lines that define the query. In the first case, all points inside the triangle are reported and in the second case the triangle is discarded, while in the third case it continues the recursion on this triangle. The number of triangles that the query can cross is bounded since each line crosses at most $O(|S|^{1/4})$ triangles at the root. The query time is $O(N^{1/2+\varepsilon})$, with the constant factor depending on the choice of ε.

Agarwal et al. [2] gave an external memory version of static partition trees that answers queries in $O(n^{1/2+\varepsilon} + k)$ I/Os. The structure can become dynamic using a standard technique by Overmars [10].It can be shown that a point is inserted or deleted in a partition tree in $O(\log_2{}^2 N)$ I/Os and answer simplex queries in $O(n^{1/2+\varepsilon} + k)$ I/O's. A method that achieves $O(\log^2_B(\frac{N}{B}))$ amortized update overhead is presented in [2].

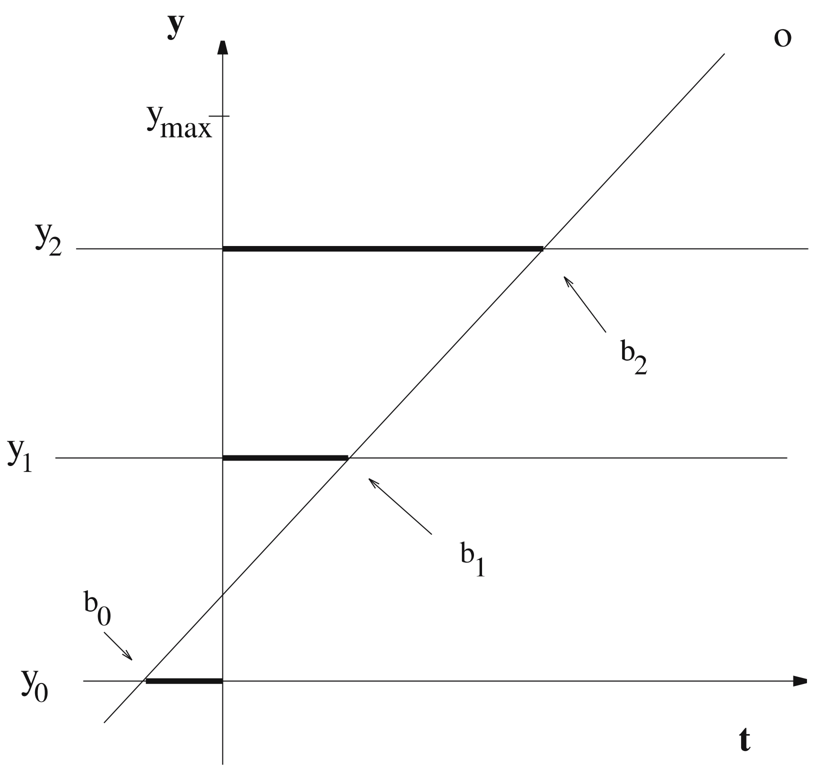
Using Point Access Methods Partition trees are not very useful in practice because the query time is $O(n^{1/2+\varepsilon} + k)$ and the hidden constant factor becomes large for small ε. In this section, two different and more practical methods are presented.

There are a large number of access methods that have been proposed to index point data. All of these structures were designed to address *orthogonal* queries, i. e., a query expressed as a multidimensional hyper-rectangle. However, most can be easily modified to address non-orthogonal queries like simplex queries.

Goldstein et al. [6] presented an algorithm to answer simplex range queries using R-trees. The idea is to change the search procedure of the tree. In particular, they gave efficient methods to test whether a linear constraint query region and a hyper-rectangle overlap. As mentioned in [6], this method is not only applicable to the R-tree family, but to other access methods as well. This approach can be used to answer the one-dimensional MOR query in the dual Hough-X space.

This approach can be improved by using a characteristic of the Hough-Y dual transformation. In this case, objects have a minimum and maximum speed, $v_{\min}$ and $v_{\max}$, respectively. The $v_{\max}$ constraint is natural in moving object databases that track physical objects. On the other hand, the $v_{\min}$ constraint comes from the fact that the Hough-Y transformation cannot represent static objects. For these objects, the Hough-X transformation is used, as it is explained above. In general, the b coordinate can be computed at different horizontal ($y = y_r$) lines. The query region is described by the intersection of two half-plane queries (Fig. 4). The first line intersects the line $n = \frac{1}{v_{\max}}$ at the point $(t_{1q} - \frac{y_{2q}-y_r}{v_{\max}}, \frac{1}{v_{\max}})$ and the line $n = \frac{1}{v_{\min}}$ at the point $(t_{1q} - \frac{y_{2q}-y_r}{v_{\min}}, \frac{1}{v_{\min}})$. Similarly, the other line that defines the query intersects the horizontal lines at $(t_{2q} - \frac{y_{1q}-y_r}{v_{\max}}, \frac{1}{v_{\max}})$ and $(t_{2q} - \frac{y_{1q}-y_r}{v_{\min}}, \frac{1}{v_{\min}})$.

Since access methods are more efficient for rectangle queries, suppose that the simplex query is approximated with a rectangular one. In Fig. 4, the query approximation rectangle will be $[(t_{1q} - \frac{y_{2q}-y_r}{v_{\min}}, t_{2q} - \frac{y_{1q}-y_r}{v_{\max}}), (\frac{1}{v_{\max}}, \frac{1}{v_{\min}})]$. Note that the query area is enlarged



Mobile Object Indexing, Figure 5
Coordinate b as seen from different 'observation' points

M

by the area $E = E^{\text{HoughY}} = E_1{}^{\text{HoughY}} + E_2{}^{\text{HoughY}}$, which is computed as:

$$E^{\text{HoughY}} = \frac{1}{2}\left(\frac{v_{\max} - v_{\min}}{v_{\min} \cdot v_{\max}}\right)^2 (|\, y_{2q} - y_r \,| + |\, y_{1q} - y_r \,|) \tag{1}$$

The objective is to minimize E since it represents a measure of the extra I/O's that an access method will have to perform for solving a one-dimensional MOR query. E is based on both y_r (i. e., where the b coordinate is computed) and the query interval (y_{1q}, y_{2q}), which is unknown. Hence, the method keeps c indices (where c is a small constant) at equidistant y_r's. All c indices contain the same information about the objects, but use different y_r's. The i-th index stores the b coordinates of the data points using $y_i = \frac{y_{\max}}{c} \cdot i, i = 0, \ldots, c-1$ (see Fig. 5). Conceptually, y_i serves as an "observation" element and its corresponding index stores the data as observed from position y_i. The area between subsequent "observation" elements is called a *sub-terrain*. A given one-dimensional MOR query will be forwarded to, and answered exactly by, the index that minimizes E.

To process a general query interval [y_{1q}, y_{2q}], two cases are considered, depending on whether the query interval covers a sub-terrain:

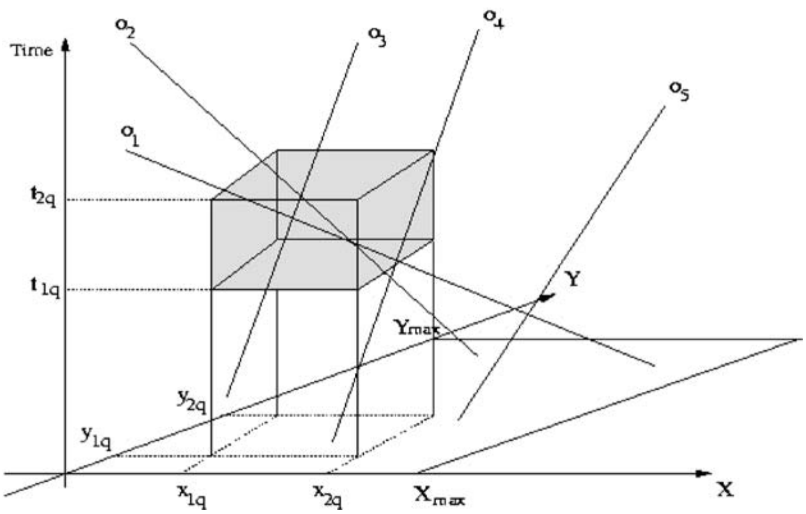
(i) $y_{2q} - y_{1q} < q\frac{y_{\max}}{c}$: then it can be easily shown that area E is bounded by:

$$E < q\frac{1}{2}\left(\frac{v_{\max} - v_{\min}}{v_{\min} \cdot v_{\max}}\right)^2 \left(\frac{y_{\max}}{c}\right). \tag{2}$$

The query is processed at the index that minimizes $|y_{2q} - y_r| + |y_{1q} - y_r|$.

(ii) $y_{2q} - y_{1q} > \frac{y_{\max}}{c}$: the query interval contains one or more sub-terrains, which implies that if a query is executed at a single observation index, area E becomes large. To bound E, index each sub-terrain too. Each of the c sub-terrain indices records the time interval when a moving object was in the sub-terrain. Then, the query is decomposed into a collection of smaller sub-queries: one sub-query per sub-terrain fully contained by the original query interval, and one sub-query for each of the original query's endpoints. The sub-queries at the endpoints fall to the case (i) above, thus, they can be answered with bounded E using an appropriate "observation" index. To index the intervals in each sub-terrain, an external memory interval tree can be used which answers a sub-terrain query optimally (i. e., $E = 0$). As a result, the original query can be answered with bounded E. However, interval trees will increase the space consumption of the indexing method.

The same approach can be used for the Hough-X transformation, where, instead of different "observation" points,



Mobile Object Indexing, Figure 6
Trajectories and query in (x,y,t) space

different "observation" times are used. That is, the intercept a can be computed using different vertical lines $t = t_i$, $i = 0, \ldots, c-1$. For each different intercept, an index is created. Then, given a query, one of the indices is chosen to answer the query (the one that is constructed for the "observation" time closest to the query time.) However, note that if the query time(s) is far from the "observation" time of an index, then the index will not be very efficient since the query in the Hough-X will not be aligned with the rectangles representing the index and the data pages of this index. Therefore, one problem with this approach comes from the fact that the time in general and the query time in particular, are always increasing. Therefore, an index that is efficient now will become inefficient later. One simple solution to this problem is to create a new index with a newer observation time every T time instants, at the same time removing the index with the oldest observation time [8,11]. Note that this problem does not exist in the Hough-Y case since the terrain and the query domain do not change with time (or they change very slowly).

Indexing in Two Dimensions

For the two-dimensional problem, trajectories of the moving objects are lines in a three dimensional space (see Fig. 6). Thus, the dual transformation gives a 4-dimensional dual point. Another approach is to split the motion of an object into two independent motions, one in the (t, x) plane and one in the (t, y) plane. Each motion is indexed separately. Next, the procedure used to build the index is presented as well as the algorithm for answering the 2-d query.

Building the Index The motion in (x, y, t) space is decomposed into two motions, one on the (t, x) and the other on the (t, y) plane. Furthermore, on each projection, the objects are partitioned according to their velocity. Objects with small velocity magnitudes are stored using the Hough-X dual transform, while the rest are stored using the Hough-Y transform, i.e, into distinct index structures.

The reason for using different transforms is that motions with small velocities in the Hough-Y approach are mapped into dual points (b, n), having large n coordinates ($n = \frac{1}{v}$). Thus, since few objects have small velocities, by storing the Hough-Y dual points in an index structure such an R*-tree, MBRs with large extents are introduced and the index performance is severely affected. On the other hand, by using a Hough-X index for the small velocities' partition, this effect is eliminated since the Hough-X dual transform maps an object's motion to the (v, a) dual point. The objects are partitioned into slow and fast using a threshold VT.

When a dual point is stored in the index responsible for the object's motion in one of the planes, i. e., (t, x) or (t,y), information about the motion in the other plane is also included. Thus, the leaves in both indices for the Hough-Y partition store the record (n_x, b_x, n_y, b_y). Similarly, for the Hough-X partition in both projections, the record (v_x, a_x, v_y, a_y) is stored. In this way, the query can be answered by one of the indices; either the one responsible for the (t, x) or the (t, y) projection.

On a given projection, the dual points (i. e., (n, b) and (v, a)) are indexed using R*-trees [3]. The R*-tree has been modified in order to store points at the leaf level and not degenerated rectangles. Therefore, extra information about the other projection can be stored. An outline of the procedure for building the index follows:

1. Decompose the 2-d motion into two 1-d motions on the (t, x) and (t, y) planes.

2. For each projection, build the corresponding index structure.

2.1 Partition the objects according to their velocity: Objects with $|v| < VT$ are stored using the Hough-X dual transform, while objects with $|v| \geq VT$ are stored using the Hough-Y dual transform.

2.2 Motion information about the other projection is also included in each point.

In order to choose one of the two projections and answer the simplex query, the following technique is used.

Answering the Query The two-dimensional MOR query is mapped to a simplex query in the dual space. The simplex query is the intersection of four 3-d hyperplanes and the projections of the query on the (t, x) and (t, y) planes are wedges, as in the one-dimensional case.

The 2-d query is decomposed into two 1-d queries, one for each projection, and it is answered exactly. Furthermore, on a given projection, the simplex query is processed in both partitions, i. e., Hough-Y and Hough-X.

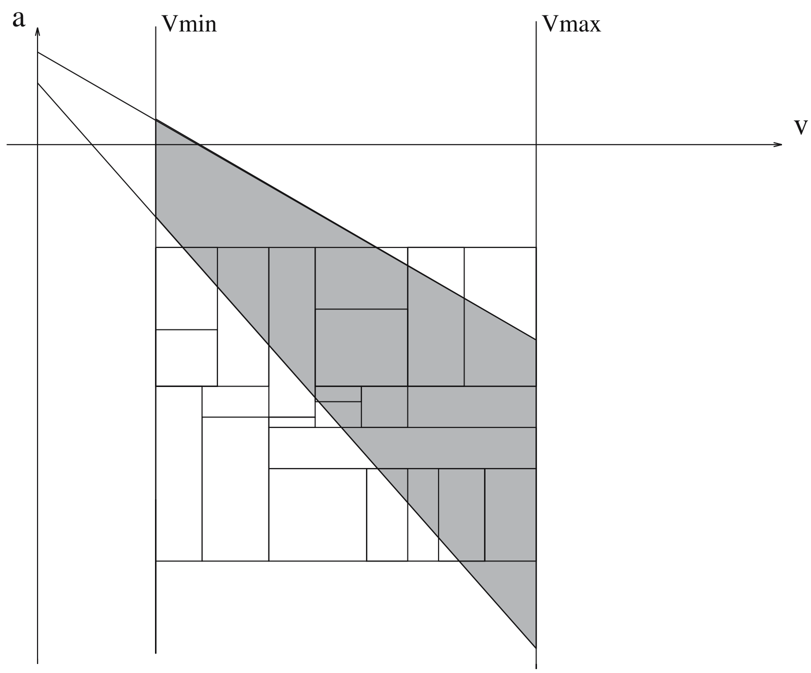
On the Hough-Y plane the query region is given by the intersection of two half-plane queries, as shown in Fig. 4. Consider the parallel lines $n = \frac{1}{v_{\min}}$ and $n = \frac{1}{v_{\max}}$. Note that a minimum value for $v_{\min}$ is VT. As illustrated in Sect. 2, if the simplex query was answered approximately, the query area would be enlarged by $E^{\text{HoughY}} = E_1{}^{\text{HoughY}} + E_2{}^{\text{HoughY}}$ (the triangular areas in Fig. 4). Also, let the actual area of the simplex query be Q^{HoughY}. Similarly, on the dual Hough-X plane (Fig. 3), let Q^{HoughX} be the actual area of the query and E^{HoughX} be the enlargement. The algorithm chooses the projection which minimizes the following criterion κ:

$$\kappa = \frac{E^{\text{HoughY}}}{Q^{\text{HoughY}}} + \frac{E^{\text{HoughX}}}{Q^{\text{HoughX}}} . \tag{3}$$

The intuition for this heuristic is that simplex queries in the dual space are not aligned with the MBRs of the underlying index (see Fig. 7). Therefore, the projection where the query is as much aligned with the MBRs as possible is chosen. The empty space, as used in the aforementioned criterion definition, gives an indication of that.

Since the whole motion information is kept in the indices, it can be used to filter out objects that do not satisfy the query. An outline of the algorithm for answering the exact 2-d query is presented below:

1. Decompose the query into two 1-d queries, for the (t, x) and (t, y) projection.
2. Get the dual query for each projection (i. e., the simplex query).
3. Calculate the criterion κ for each projection and choose the one (say p) that minimizes it.
4. Answer the query by searching the Hough-X and Hough-Y partition using projection p.
5. Put an object in the result set only if it satisfies the query. Use the whole motion information to do the filtering “on the fly”.



Mobile Object Indexing, Figure 7 Simplex query in dual space, not aligned with MBRs of underlying index

Key Applications

Location-aware applications such as traffic monitoring, intelligent navigation, and mobile communications management require the storage and retrieval of the locations of continuously moving objects. For example, in a company that manages taxi services, a customer may want to find the taxis that will be in a specific area in the near future. This can be achieved by issuing a query: "Report the taxis that will be in the area around the customer in the next 5 minutes". Similarly, in an air-traffic control system, the locations of the airplanes that are flying close to an airport or a city must be continuously monitored. In that case, a predictive range query can be periodically issued every few seconds. The index will speed up the search and allow for multiple queries to be executed each minute.

Future Directions

There are a number of interesting future problems related to the discussed topic. The dual transformation has been used for objects moving in one and two dimensions. It is interesting to investigate how these methods will be extended to three or higher dimensions. Although the described methods can be applied on higher dimensions, it is not clear if they will have the same efficiency and practicality. Most of the work has been done for linearly moving objects. An interesting future direction is to consider storage and retrieval of non-linear movements. Another problem is to consider moving objects with extents that change over time in addition to their location.

Recommended Reading

1. Agarwal, P., Har-Peled, S.: Maintaining Approximate Exten Measures of Moving Points. In: Proceedings of the 12th ACM-SIAM Sympos. Discrete Algorithms, Washington, DC, USA, 7–9 Jan 2001, pp. 148–157
2. Agarwal, P.K., Arge, L., Erickson, J.: Indexing Moving Points. In: Proceedings of the 19th ACM Symp. on Principles of Database Systems, Dallas, Texas, USA, 15–17 May 2000, pp. 175–186
3. Beckmann, N., Kriegel, H., Schneider, R., Seeger, B.: The R*-tree: An Efficient and Robust Access Method for Points and Rectangles. In: Proceedings of the 1990 ACM SIGMOD, Atlantic City, May 1998, pp. 322–331
4. Chon, H.D., Agrawal, D., Abbadi, A.E.: Query Processing for Moving Objects with Space-Time Grid Storage Model. In: Proceedings of the 3rd Int. Conf. on Mobile Data Management, Singapore, 8–11 Jan 2002, pp. 121–126
5. Elbassioni, K., Elmasry, A., Kamel, I.: An efficient indexing scheme for multi-dimensional moving objects. In: Proceedings of the 9th Intern. Conf. ICDT, Siena, Italy, 8–10 2003, pp. 425–439
6. Goldstein, J., Ramakrishnan, R., Shaft, U., Yu, J.: Processing Queries By Linear Constraints. In: Proceedings of the 16th ACM PODS Symposium on Principles of Database Systems, Tucson, Arizona, USA, 13–15 May 1997, pp. 257–267
7. Jensen, C.S., Lin, D., Ooi, B.C.: Query and Update Efficient B^+-Tree Based Indexing of Moving Objects. In: VLDB, Toronto, Canada, 29 Aug – 3 Sept 2004, pp. 768–779
8. Kollios, G., Gunopulos, D., Tsotras, V.: On Indexing Mobile Objects. In: Proceedings of the 18th ACM Symp. on Principles of Database Systems, Philadelphia, Pennsylvania, USA, 1–3 June 1999, pp. 261–272
9. Matousek, J.: Efficient Partition Trees. Discret. Comput. Geom. **8**, 432–448 (1992)
10. Overmars, M.H.: The Design of Dynamic Data Strucutures. vol. 156 of LNCS. Springer-Verlag, Heidelberg (1983)
11. Patel, J., Chen, Y., Chakka, V.: STRIPES: An Efficient Index for Predicted Trajectories. In: Proceedings of the 2004 ACM SIGMOD, Paris, France, 13–18 June 2004, pp. 637–646
12. Procopiuc, C.M., Agarwal, P.K., Har-Peled, S.: Star-tree: An efficient self-adjusting index for moving objects. In: Proceedings of the 4th Workshop on Algorithm Engineering and Experiments, San Francisco, CA, USA, 4–5 Jan 2002, pp. 178–193
13. Saltenis, S., Jensen, C., Leutenegger, S., Lopez, M.A.: Indexing the Positions of Continuously Moving Objects. In: Proceedings of the 2000 ACM SIGMOD, Dallas, Texas, USA, 16–18 May 2000, pp. 331–342
14. Tao, Y., Papadias, D., Sun, J.: The TPR*-Tree: An Optimized Spatio-temporal Access Method for Predictive Queries. In: Proceedings of the 29th Intern. Coonf. on Very Large Data Bases, Berlin, Germany, 9–12 Sept 2003, pp. 790–801
15. Tayeb, J., Olusoy, O., Wolfson, O.: A Quadtree-Based Dynamic Attribute Indexing Method. Comp. J., **41**(3), 185–200 (1998)
16. Wolfson, O., Xu, B., Chamberlain, S., Jiang, L.: Moving Objects Databases: Issues and Solutions. In: Proceedings of the 10th Int. Conf. on Scientfic and Statistical Database Management, Capri, Italy, 1–3 July 1998, pp. 111–122
17. Yiu, M.L., Tao, Y., Mamoulis, N.: The B^{dual}-Tree: Indexing Moving Objects by Space-Filling Curves in the Dual Space. VLDB J., To Appear

Mobile Objects Databases

HARVEY J. MILLER
Department of Geography, University of Utah,
Salt Lake City, UT, USA

Synonyms

Moving objects database

Definition

Mobile objects databases (MOD) store data about entities that can change their geometry frequently, include changes in location, sizes and shapes

Main Text

Most database management systems (DBMS), even spatio-temporal DBMS, are not well equipped to handle data on objects that change their geometry frequently, in some cases continuously. In DBMS, data is assumed to be constant unless it is explicitly modified. Using standard database update techniques to update the geometry of a moving object is too expensive computationally. Traditional query languages such as structured query language (SQL) are not well-equipped to handle the spatio-temporal queries required by a MOD. These include: "Retrieve all objects that will intersect a region within the next 4 minutes." "Retrieve all objects that will come within 3 kilometers of each other and the time when this will occur." Finally, although the geometry of a moving object is changing continuously in some cases, digital technology for recording these geometries (such as location-aware technologies) as well as technologies for storing these data have finite resolution: it can only record and store position at discrete moments. Also, each one of these positions will have a degree of imprecision. Therefore, the position of an object at any given moment in time will have a degree of uncertainty.

MODs handle all three components of data that change their geometry frequently, namely: i) rapid database updating; ii) mobile object queries; iii) uncertainty in locational tracking.

Although MODs theoretically handle entities that change their location, size and shapes, many applications focus on rigid objects that only change their positions such as vehicles.

Cross References

► Geographic Knowledge Discovery
► Location-Aware Technologies

Recommended Reading

1. Güting, R.H., Schneider, M.: Moving Objects Databases. Morgan Kaufmann, San Francisco, CA, USA (2005)
2. Wolfson, O., Xu, B., Chamberlain, S., Jiang, L.: Moving objects databases: Issues and solutions. In: Proceedings of the 10th International Conference on Scientific and Statistical Database Management (SSDBM98) pp. 111–122 (1998)

Mobile P2P Databases[1]

YAN LUO, OURI WOLFSON
Department of Computer Science,
University of Illinois at Chicago, Chicago, IL, USA

Definition

A mobile peer-to-peer (P2P) database is a database that is stored in the peers of a mobile P2P network. The network is composed by a finite set of mobile peers that communicate with each other via short-range wireless protocols, such as IEEE 802.11, Bluetooth, Zigbee, or ultra wide band (UWB). These protocols provide broadband (typically tens of Mbps) but short-range (typically 10–100 meters) wireless communication. On each mobile peer there is a local database that stores and manages a collection of data items or reports. A report is a set of values sensed or entered by the user at a particular time, or otherwise obtained by a mobile peer. Often a report describes a physical resource, such as an available parking slot. All the local databases maintained by the mobile peers form the mobile P2P database. The peers communicate reports and queries to neighbors directly, and the reports and queries propagate by transitive multi-hop transmissions. Figure 1 below illustrates the definition.

In contrast to the assumptions made in the literature on mobile ad hoc networks (MANETs) and mesh networks, a peer may not know the identities of other peers in the network and the data they store. Thus, routing in the traditional MANET sense is not a common operation in mobile P2P databases.

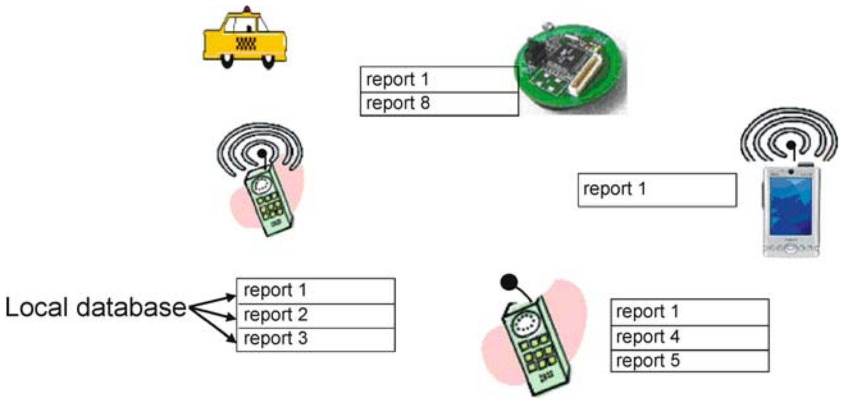
Mobile P2P databases enable matchmaking or resource discovery services in many application domains, including social networks, transportation, mobile electronic commerce, emergency response, and homeland security.

Communication is often restricted by bandwidth and power constraints on the mobile peers. Furthermore, often reports need to be stored and later forwarded, thus memory constraints on the mobile devices constitute a problem as well. Thus, careful and efficient utilization of scarce peer resources (specifically bandwidth, power, and memory) are an important challenge for mobile P2P databases.

Historical Background

Traditionally search databases have been implemented by a centralized architecture. Google is preeminent example of such architecture. However, mobile P2P databases have several advantages over centralized ones. First, because short-range wireless networks utilize the unlicensed spectrum, communication to the mobile P2P database is free; there is also no cost involved in setting up and maintaining the fixed infrastructure database. Second, mobile P2P databases can be used for search in emergency, disaster, and other situations where the infrastructure is destroyed or unavailable. Third, mobile P2P databases are harder to mine for private information, and fourth, they are more reliable in the sense that failure of the central site will not render the system unavailable. Fifth, mobile P2P databases can withstand the high update rates that will be generated

[1]Research supported by NASA Award NNA06AA25C, and NSF Grants OII-0611017, DGE-0549489, 0513736, and 0326284.



Mobile P2P Databases, Figure 1
A mobile P2P database

when representing temporary physical resources (e. g. the available parking slots), or continuous phenomena, such as the location of moving objects. The disadvantage of mobile P2P databases is that they do not provide answer guarantees. In other words, although the answer to a query exists in the database, due to mobility and lack of global coordination, the mobile P2P database may not find it.

The concept of mobile P2P database is proposed for searching local information, particularly information of a temporary nature, i. e., valid for a short duration of time [1].

Currently, there are quite a few experimental projects in mobile P2P databases. These can be roughly classified into pedestrians and vehicular projects. Vehicular projects deal with high mobility and high communication topology change-rates, whereas pedestrian projects have a strong concern with power issues. The following are several active experimental mobile P2P database projects for pedestrians and vehicles:

Pedestrians Projects

- **7DS** – Columbia University ([3])
  - http://www.cs.unc.edu/~maria/7ds/
  - Focuses on accessing web pages in environments where only some peers have access to the fixed infrastructure.
- **iClouds** – Darmstadt University ([4])
  - http://iclouds.tk.informatik.tu-darmstadt.de/
  - Focuses on the provision of incentives to brokers (intermediaries) to participate in the mobile P2P database.
- **MoGATU** – University of Maryland, Baltimore County ([5])
  - http://mogatu.umbc.edu/
  - Focuses on the processing of complex data management operations, such as joins, in a collaborative fashion.
- **PeopleNet** – National University of Singapore ([6])
  - http://www.ece.nus.edu.sg/research/projects/abstract.asp?Prj=101
  - Proposes the concept of information bazaars, each of which specializes in a particular type of information; reports and queries are propagated to the appropriate bazaar by the fixed infrastructure.
- **MoB** – University of Wisconsin and Cambridge University ([7])
  - http://www.cs.wisc.edu/~suman/projects/agora/
  - Focuses on incentives and the sharing among peers of virtual information resources such as bandwidth.
- **Mobi-Dik** – University of Illinois at Chicago ([1,2])
  - http://www.cs.uic.edu/~wolfson/html/p2p.html
  - Focuses on information representing physical resources, and proposes stateless algorithms for query processing, with particular concerns for power, bandwidth, and memory constraints.

Vehicular Projects

- **CarTALK 2000** – A European project
  - http://www.cartalk2000.net/
  - Develops a co-operative driver assistance system based upon inter-vehicle communication and mobile P2P databases via self-organizing vehicular ad hoc networks.
- **FleetNet** – Internet on the Road Project ([8])
  - http://www.ccrle.nec.de/Projects/fleetnet.htm
  - Develops a wireless multi-hop ad hoc network for intervehicle communication to improve the driver's and passengers' safety and comfort. A data dissemination method called "contention-based forwarding" (CBF) is proposed in which the next hop in the forwarding process is selected through a distributed contention mechanism based on the current positions of neighbors.

- **VII** – Vehicle Infrastructure Integration, a US DOT project
  - http://www.its.dot.gov/vii/
  - The objective of the project is to deploy advanced vehicle-to-vehicle (using the mobile P2P paradigm) and vehicle-to-infrastructure communications that could keep vehicles from leaving the road and enhance their safe movement through intersections.
- **Grassroots** – Rutgers University ([9])
  - http://paul.rutgers.edu/~gsamir/dataspace/grassroots.html
  - Develops an environment in which each vehicle contributes a small piece of traffic information to the network based on the P2P paradigm, and each vehicle aggregates pieces of the information into a useful picture of the local traffic information.

Scientific Fundamentals

There are two main paradigms for answering queries in mobile P2P databases, one is report pulling and the other one is report pushing.

Report pulling means that a mobile peer makes an explicit request for the report it is interested in receiving, and the whole network is flooded with queries, the interested report will be pulled from the mobile peers that have them. Report pulling is widely used in resource discovery, such as route discovery in mobile ad hoc networks and file discovery by query flooding in wired P2P networks like Gnutella. Flooding in a wireless network is in fact relatively efficient as compared to wired networks because of wireless multicast advantage.

Another possible approach for data dissemination is report pushing. Report pushing is the dual problem of report pulling; reports are flooded, and consumed by peers whose query is answered by received reports. So far there exist mechanisms to broadcast information in the complete network, or in a specific geographic area (geocast), apart from to any one specific mobile node (unicast/mobile ad hoc routing) or any one arbitrary node (anycast). Report pushing paradigm can be further divided into stateful methods and stateless methods. Most stateful methods are topology-based, i. e., they impose a structure of links in the network, and maintain states of data dissemination. PStree, which organizes the peers as a tree, is an example of topology-based methods.

Another group of stateful methods is the cluster- or hierarchy-based method, such as [13], in which moving peers are grouped into some clusters or hierarchies and the cluster heads are randomly selected. Reports are disseminated through the network in a cluster or hierarchy manner, which means that reports are first disseminated to every cluster head and each cluster head then broadcasts the reports to the member peers in its group. Although cluster- or hierarchy-based methods can minimize the energy dissipation in moving peers, these methods will fail or cost more energy in highly mobile environments since they have to maintain a hierarchy structure and frequently reselect cluster heads.

Another stateful paradigm consists of location-based methods (see [14]). In location-based methods, each moving peer knows the location of itself and its neighbors through some localization techniques, such as GPS or atomic multilateration (see [14]).

The simplest location-based data dissemination is greedy forwarding, in which each moving peer transmits a report to a neighbor that is closer to the destination than it is. However, greedy forwarding can fail in some cases, such as when a report is stuck in local minima, which means that the report stays in a mobile peer whose neighbors are all further from the destination. Therefore, some recovery strategies are proposed, such as greedy perimeter stateless routing (GPSR) [15]. Other location-based methods, such as geographic adaptive fidelity (GAF) [16] and geographical and energy aware routing (GEAR) [17]), take advantage of knowledge about both location and energy to disseminate information and resources more efficiently.

In stateless methods, the most basic and simplest one is the flooding-based method, such as [10]. In flooding-based methods, mobile peers simply propagate received reports to all neighboring mobile peers until the destination or maximum a hop is reached. Each report is propagated as soon as is received. Flooding-based methods have many advantages, such as no state maintenance, no route discovery, and easy deployment. However, they inherently cannot overcome several problems, such as implosion, overlap, and resource blindness. Therefore, other stateless methods are proposed, such as gossiping-based methods and negotiation-based methods.

Gossiping-based methods, such as [11], improve flooding-based methods by transmitting received reports to a randomly selected neighbor or to the neighbors that are interested in the particular content. The advantages of gossiping-based methods include reducing the implosion and lowering the system overhead. However, the cost of determining the particular interests of each moving peer can be huge and transmitting reports to a randomly selected neighbor can still cause the implosion problem and waste peers' memory, bandwidth and energy. Furthermore, dissemination and, thus, performance are reduced compared to pure flooding.

Negotiation-based methods solve the implosion and overlap problem by transmitting first the IDs of reports; the reports themselves are transmitted only when requested (see [12]). Thus, some extra data transmission is involved, which costs more memory, bandwidth, and energy. In addition, in negotiation-based methods, moving peers have to generate meta-data or a signature for every report so that negotiation can be carried out, which will increase the system overhead and decrease the efficiency.

Another important stateless paradigm for data dissemination in mobile P2P networks is store-and-forward, such as [2], which to ranks all the reports in a peer's database in terms of their relevance or expected utility, and then the reports are communicated and saved in the order of their relevance. Alternatively, the reports requested and communicated are the ones with the relevance above a certain threshold. The notion of relevance quantifies the importance or the expected utility of a report to a peer at a particular time and at a particular location. Other store-and-forward methods include PeopleNet [6] and 7DS [3].

In summary, the paradigms for data dissemination in mobile P2P databases are summarized in Fig. 2 below.

Key Applications

Mobile P2P databases provide mobile users a search engine for transient and highly dynamic information in a local geospatial environment. Mobile P2P databases employ a unified model for both the cellular infrastructure and the mobile ad hoc environments. When the infrastructure is available, it can be augmented by the mobile P2P database approach.

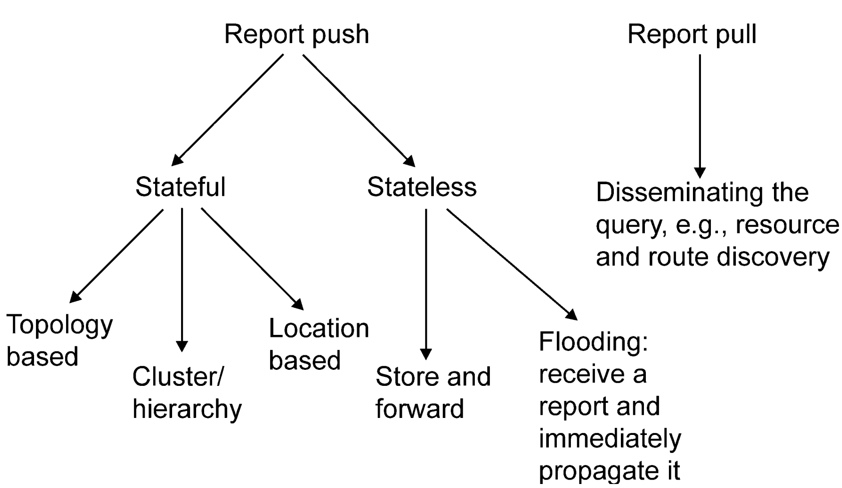
Consider a mobile P2P database platform, i. e., a set of software services for data management in a mobile P2P environment; it is similar to a regular database management system, but geared to mobile P2P interactions. Such a platform will enable quick building of matchmaking or resource discovery services in many application domains, including social networks, emergency response, homeland security, military, airport applications, mobile e-commerce, and transportation.

Social Networks

In a large professional, political, or social gathering, mobile P2P databases are useful to automatically facilitate a face-to-face meeting based on matching profiles. For example, in a professional gathering, mobile P2P databases enable attendees to specify queries (interest profiles) and resource descriptions (expertise) to facilitate conversations, when mutual interest is detected. This opportunistic matchmaking can greatly enhance the value of networking events allowing users to connect with targeted, interested parties without a priori knowledge of their name, title, phone number, or other personal information. A face-to-face meeting can be setup by including in the resource description the identification information of the resource (person), such as cell-phone number, name, screen name, picture, physical description, etc. This information may be used together with the (possibly imprecise) location to help set up the face-to-face meeting. Thus, the individual's profile that is stored in mobile P2P databases will serve as a "wearable web-site". Similarly, mobile P2P databases can facilitate face-to-face meetings in singles matchmaking.

Emergency Response, Homeland Security, and the Military

Mobile P2P databases offer the capability to extend decision-making and coordination capability. This finds applications in emergency environments, an area of particular concern to the government trying to find technologies that can be exploited to support the more than eight million first responders in US homeland security. Consider workers in disaster areas, soldiers and military personnel oper-



Mobile P2P Databases, Figure 2 Query answering methods in mobile P2P databases ating in environments where the wireless fixed infrastructure is significantly degraded or non-existent. They would welcome a capability that lets them automatically propagate messages, pictures, or resource information to other workers, based on matching profiles, security, and attribute values rather than node-id. As mobile users involved in an emergency response naturally cluster around the location of interest, a self-forming, high-bandwidth network that allows secure point-to-point or point-to-multipoint communication without the need of potentially compromised infrastructure could be of great benefit. For instance, a picture of a wanted person could be propagated to all those involved in a targeted search at the scene.

Consider a related emergency response application. Scientists are developing cockroach-sized robots or sensors that are carried by real cockroaches, which are able to search victims in exploded or earthquake-damaged buildings. These robots or sensors are equipped with radio transmitters. When a robot discovers a victim by sensing carbon dioxide, it may not have the transmission power to reach the outside rescuers; it can use local data dissemination to propagate the information to human rescuers outside the rubble. Sensors can also be installed on wild animals for endangered species assistance. A sensor monitors its carrier's health condition, and it disseminates a report when an emergency symptom is detected.

Airport Applications

Airports provide several different opportunities for the use of mobile P2P databases. From the point of view of commerce, airports have stores and kiosks where merchandise is sold similarly to a mall. Imagine arriving at a large airport and realizing you do not have the computer power cord you need for your presentation. Mobile P2P databases will enable a user to search for the needed product - just like in a mall. Merchants can similarly provide their location information and offer promotional incentives to passengers.

Mobile P2P databases can also be used by airport personnel to coordinate their activities. This is especially important when there is a communication failure due an emergency that degrades the infrastructure. Like the case of early responders, airport personnel can continue to coordinate their activities through the use of the mobile P2P network that is available even though the infrastructure is not functioning. Another potential opportunity that will benefit both the consumer and the airport operations is the dissemination of real-time information regarding flight changes, delays, queue length, parking information, special security alerts and procedures, and baggage information. This can augment the present audio announcements that often cannot be heard in nearby restaurants, stores, or restrooms, and the limited, expensive displays.

Mobile E-commerce

Consider short-range wireless broadcast and mobile P2P dissemination of a merchant's sale and inventory information. It will enable a customer (whose cell phone is mobile P2P databases enabled) that enters a mall to locate a desired product at the best price. When a significant percentage of people have mobile devices that can query retail data, merchants will be motivated to provide inventory/sale/coupons information electronically to nearby potential customers. The information will be provided and disseminated in a P2P fashion (in, say, a mall or airport) by the mobile P2P databases software.

Transportation Safety and Efficiency

Mobile P2P databases software can improve safety and mobility by enabling travelers to cooperate intelligently and automatically. A vehicle will be able to automatically and transitively communicate to trailing vehicles its "slow speed" message when it encounters an accident, congestion, or dangerous road surface conditions. This will allow other drivers to make decisions, such as finding alternative roads. Also, early warning messages may allow a following vehicle to anticipate sudden braking or a malfunctioning brake light, thus preventing pile-ups in some situations. Similarly, other resource information, such as ridesharing opportunities, transfer protection (transfer bus requested to wait for passengers), will be propagated transitively, improving the efficiency of the transportation system.

Inefficiencies in the transportation system result in excessive environmental pollution, fuel consumption, risk to public safety, and congestion. Ridesharing (i. e., vehicles carrying more than one person, either publicly provided such as transit, a taxi, or a vanpool, or prearranged rides in a privately owned vehicle) and car sharing (i. e., a program that allows registered users to borrow a car on an hourly basis from fixed locations) have the potential to alleviate these problems. Currently the matchmaking required in ridesharing is performed offline. However, the success of ridesharing will depend largely on the efficient identification and matching of riders/drivers to vehicles in real time in a local environment, which is where the benefit of our technology lies, providing information that is simultaneously relevant in time, location, and interest. Mobile P2P databases incorporated in navigational devices and PDA's can be used to disseminate to other devices and PDA's information about relevant resources, such as ridesharing partners, free parking slots, and available taxicabs or taxicab customers.

Future Directions

There are many challenges and directions for the future research in mobile P2P databases in mobile P2P networks:

1. **Prolong network lifetime**
   How to maximize the network life is a common but difficult problem in mobile P2P databases. Currently, some approaches as discussed above, e. g., ranking and cluster-based-methods, are proposed to address this problem and prolong the lifetime of sensor networks, mobile ad hoc networks, and mobile P2P databases. The future research question is how to employ the redundancy of networks and the density of peers in order to maximally extend the network lifetime.
2. **Sparse networks**
   Currently, the performance of many algorithms and systems heavily depends on the density of peers in mobile P2P networks. They do not perform very well if the network is sparse. Therefore, understanding how to design and develop mobile P2P databases for sparse networks is an important and difficult challenge. Recent work that heads in this direction includes delay tolerant networks, store and forward flooding, and mobile peers whose sole function is to provide connectivity.
3. **Rapid topology changes**
   Another challenge for designing and developing mobile P2P databases is high mobility of peers. This poses problems to mobile P2P databases, e. g., how to efficiently disseminate queries and answers, and how to reconfigure rapidly when the topology of networks changes frequently. Stateless approaches seem most suitable to address these problems.
4. **Emergent global behavior from local knowledge**
   Mobile P2P databases can be treated as a special type of distributed system. Each peer maintains a local database and all the local databases form the virtual mobile P2P database. Therefore, peers can only use the local knowledge to predict or affect the global behavior of the whole mobile P2P database. The future research direction will be how to employ the local knowledge and propose the adaptive local algorithms to direct or affect the global behavior of mobile P2P databases.
5. **(Self-) localization techniques**
   Location-based approaches are more and more popular and necessary, and location information of peers is useful for efficiently storing and managing information. However, self-localization techniques are still not efficient and effective enough due to the limitation of peers or localization techniques. For example, GPS is not available indoors and the accuracy of GPS is not enough for some mobile P2P databases. Therefore, creating efficient and effective self-localization technique for mobile P2P databases is an important research direction.
6. **Integration of mobile P2P databases and infrastructure**
   As discussed above, mobile P2P databases do not guarantee answer completeness. In this sense, the integration with an available infrastructure, such as the Internet or a cellular network may improve performance significantly. This integration has two aspects. First, using the communication infrastructure in order to process queries more efficiently in the mobile P2P database; and second, using data on the fixed network in order to provide better and more answers to a query. The seamless integration of mobile P2P databases and infrastructure databases introduces important research challenges. Recent work on data integration in the database community can provide a starting point for is research.
7. **Specialized queries**
   Existing mobile P2P query processing methods deal with simple queries, e. g. selections; each query is satisfied by one or more reports. However, in many application classes one may be interested in more sophisticated queries. For example, in mobile electronic commerce a user may be interested in the minimum gas price within the next 30 miles on the highway. Processing of such queries may present interesting optimization opportunities.
8. **Mathematical modeling of data dissemination**
   Many query processing and data dissemination algorithms may benefit from a mathematical model of data propagation. For example, a formula giving the number n of mobile peers having a report that was generated at time t at location l would be very useful in ranking of such a report. The number n is a function of the density of mobile peers, motion speed, bandwidth and memory availability at the peers, memory management, etc. Related work done in epidemiology about the spread of infectious diseases would be a good starting point for this research. Results in random graphs may also be applicable.

Other important research directions include incentives for broker participation in query processing, and transactions/atomicity/recovery issues in databases distributed over mobile peers.

Recommended Reading

1. Xu, B., Wolfson, O.: Data Management in Mobile P2P Networks, Springer Verlag Lecture Notes in Computer Science, Proc. of the 2nd International Workshop on Databases, Information Systems, and P2P Computing (DBISP2P'04), Toronto, Canada, Aug 2004

2. Wolfson, O., Xu, B., Yin, H.B., Cao, H.: Search-and-discover in mobile P2P network databases. In Proc. of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS'06), Lisbon, Portugal (2006)
3. Papadopouli, M., Schulzrinne, H.: Design and implementation of a P2P data dissemination and prefetching tool for mobile users, First NY Metro Area Networking Workshop, IBM TJ Watson Research Center, Hawthorne, NY (2001)
4. Heinemann, A., Kangasharju, J., Lyardet, F., Mühlhäuser, M.: iClouds – P2P information sharing in mobile environments, International Conference on Parallel and Distributed Computing (Euro-Par 2003), Klagenfurt, Austria (2003)
5. Perich, F.: On P2P Data Management in Pervasive Computing Environments, Ph. D. thesis, UMBC, May 2004
6. Motani, M., Srinivasan, V., Nuggehalli, P.: PeopleNet: engineering a wireless virtual social network, International Conference Mobile Computing and Networking (MobiCom'05), Aug 2005
7. Chakravorty, R., Agarwal, S., Banerjee, S., Pratt, I.: MoB: A mobile bazaar for wide-area wireless services, International Conference Mobile Computing and Networking (MobiCom'05), Aug 2005
8. Franz, W., Eberhardt, R., Luckenbach, T.: FleetNet – Internet on the road, The 8th World Congress on Intelligent Transportation Systems, Sydney, Australia, Oct 2001
9. Goel, S., Imielinski, T., Ozbay, K., Nath, B.: Grassroots: A scalable and robust information architecture, Technical Report DCS-TR-523, Department of Computer Science, Rutgers University, June 2003
10. Oliveira, R., Bernardo, L., Pinto, P.: Flooding techniques for resource discovery on high mobility MANETs, Workshop on Wireless ad hoc Networks (2005)
11. Datta, A., Quarteroni, S., Aberer, K.: Autonomous Gossiping: A self-Organizing epidemic algorithm for selective information dissemination in wireless mobile ad hoc networks. In: The International Conference on Semantics of a Networked World (2004)
12. Kulik, J., Heinzelman, W., Balakrishnan, H.: Negotiation-based protocols for disseminating information in wireless sensor networks, Wireless Networks, vol. 8, pp. 169–185 (2002)
13. Visvanathan, A.,Youn, J.H., Deogun, J.: Hierarchical data dissemination scheme for large scale sensor networks. In: IEEE International Conference on Communications (ICC'05), pp. 3030–3036, May 2005
14. Mauve, M., Widmer, A., Hartenstein, H.: A survey on position-based routing in mobile ad hoc networks, IEEE Network, vol. 15, no. 6, pp. 30–39 (2001)
15. Karp, B., Kung, H.T.: GPSR: Greedy perimeter stateless routing for wireless sensor networks. In: The 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00), pp. 243–254, Aug 2000
16. Xu, Y., Heidemann, J., Estrin, D.: Geography-informed energy conservation for ad hoc routing. In: The ACM International Conference on Mobile Computing and Networking, pp. 70–84, Rome, Italy, July 2001
17. Yu, Y., Govindan, R., Estrin, D.: Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks, Technical Report UCLA/CSD-TR-01-0023,UCLA, May 2001

Mobile Population

► Population Distribution During the Day

Mobile Robotics

► Indoor Positioning, Bayesian Methods

Mobile Usage

► Mobile Usage and Adaptive Visualization

Mobile Usage and Adaptive Visualization

TUMASCH REICHENBACHER
Department of Geography, University of Zurich,
Zurich, Switzerland

Synonyms

Personalized visualization; Personalized maps; Customization; Adaption, complete; Adaptive, context-aware

Definition

Generally, adaptive visualization is the adjustment of the visualization of geographic information and associated parts in the visualization process such as the interface, the information content, and the information encoding by a visualization application or a geospatial web service to a specific usage context.

The concept of adaptation has been applied mainly to the mobile usage context, where maps and other visualization forms of geographic information are stored and/or displayed on portable devices that are in most cases owned by the user and are constantly carried around. In such a context, the mobile user generally demands geospatial information that is retrieved either locally or remotely over an internet connection and visualized through the mediation of a user interface [1]. Figure 1 outlines the main components of the adaptation of visualization. First of all, the reason for the adaptation is given through the need for improving the usability of the mobile geospatial information access, limitations of resources in the mobile usage context, and the desire to enhance the relevance of the presented geospatial information. The adaptation concept further distinguishes what is adapted, the so-called adaptee, from what it is adapted to, the adaptation target. The former are the objects of a visualization of geographic information that are adaptable and the latter is the referential source of information to which the adaptation is directed to, i. e., the mobile usage context. Dependent on changes or differences in the usage context, adaptation methods are triggered that adapt the adaptation objects to the adaptation target.

Mobile Usage and Adaptive Visualization, Figure 1 The basic components of adaptive visualization (based on [5])

The adaptation target, the mobile usage context, and its components will be analyzed in more detail below.

Historical Background

The first attempts at adaptivity were made in the field of human–computer interaction in the late 1980s. Numerous prototypes of adaptive user interfaces were developed documented in the literature of the early 1990s [2,3,4]. The rise of multimedia provided the next field of adaptivity. In the latter half of the 1990s, researchers focused more on adaptive hypermedia [5]. The first investigation of adapting maps and the visualization of geographic information for mobile usage dates roughly to the new millennium when mobile network technology had matured and mobile telecommunications had become a mass market. This new technology, based on cell nets and mobile phones, allowed users to access geographic information almost everywhere and revealed at the same time the need for adaptation. Around the same time the first location-based services (LBS) were developed. This concept is related to adaptive visualization of geographic information, although it is much narrower and more technology-oriented as elaborated below.

Scientific Fundamentals

The prerequisite of any change to visualization is that the visualization is in itself flexible and could principally be changed at all. This quality refers to the term adaptability. In the case of geospatial visualization this has only become possible with digital representations of geographic information. Digital representations of geographic information separate the storage and the display of the information offering the flexibility for any possible changes or adjustments that were not possible with analogue paper maps or atlases. The latter are not adaptable and hence not susceptible to adaptation.

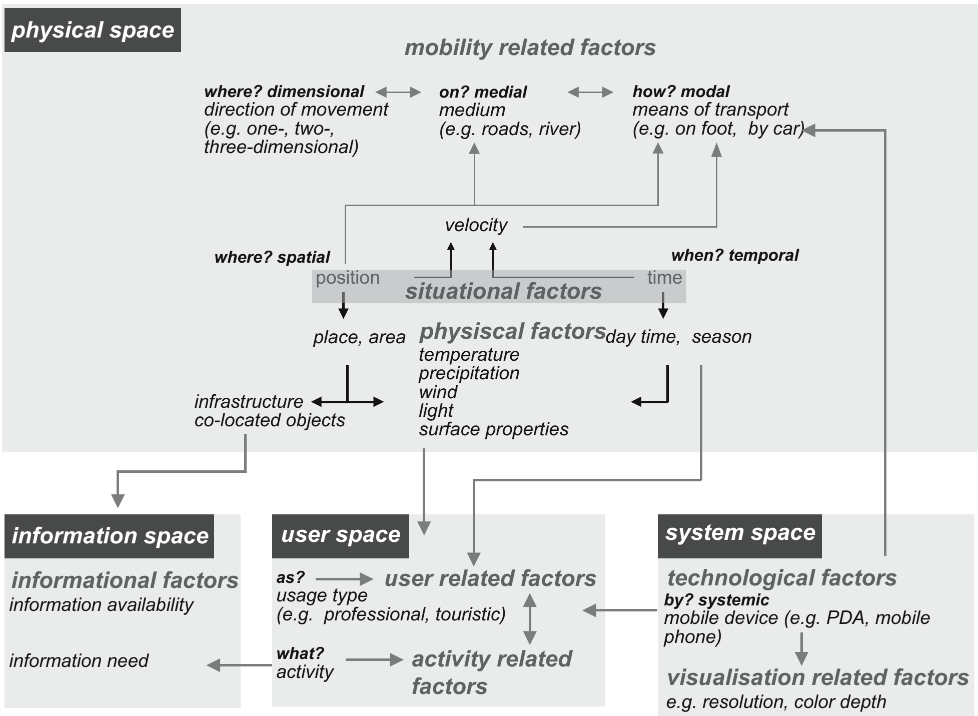
Regarding the adaptation of geospatial visualizations, two types can be distinguished. Adaptable geospatial visualizations offer the user tools to change and modify properties of the visualization. This corresponds basically to the concept of customization.

Adaptive geospatial visualizations, on the other hand, can change their characteristics automatically based on the usage context. Often this adaptivity is called "self-adapting". This dichotomy of adaptable and adaptive systems is depicted in Fig. 2 [3].

Adaptivity can take many forms depending on whether and how much the user and the system are involved in initiating, deciding about and conducting adaptation steps. For a detailed analysis of possible combinations of adaptivity, see [4]. If the focus of the adaptation is on the user, often the terms individualization or personalization are used. A good overview on the general topic of adaptation is given in [2,3].

The prerequisite for the mobile usage of geographic information is remote access to geospatial information. This is provided by information transmission through mobile telecommunication networks and the availability of small, portable computing devices. These two major technological fields, mobile telecommunication and mobile computing, offer the user the mobility of geospatial information

Mobile Usage and Adaptive Visualization, Figure 2 The adaptivity–adaptability dichotomy



Mobile Usage and Adaptive Visualization, Figure 3 The dimensions of the mobile usage context

usage. This mobility of the user and hence the usage of the geographic information lead to different and changing usage contexts.

In the definition of adaptive visualization, it was stated that the target of the adaptation is the mobile usage context. This context is naturally composed of several dimensions. Figure 3 illustrates these mobile usage context dimensions, some of their interrelationships and a few exemplary parameters for these dimensions. First of all it is possible to distinguish the scope of the context: the information space, the physical space, the visualization space, and the user space. These spaces are characterized by different contextual factors.

The physical space is primarily defined by the position of the user in space and time. These two parameters define the user's situation. Any situation is characterized by further physical conditions such as light conditions, temperature, precipitation, surface etc. Some of these parameters also influence the user's mobility which is typified by mobility factors such as the direction, the medium and mode of the movement. The mobility factors, as well as the situation, have themselves a strong influence on the user's activity. Space, time, and mobility constrain the user activities. The user space incorporates factors related to the user characteristics, the kind of geographic information usage, and user activities. Some adaptation approaches regard the user as a separate source of information for the adaptation process. This is for example the case in adaptive user interfaces or personalization of maps, where the focus is more on modeling the user and his or her characteristics. However, it is argued that the user is a central part of the mobile usage context and therefore better to be modeled together with the other context dimensions. The activities of a mobile user have an influence on the informational factors, e. g., the information needed for the successful accomplishment of current actions to reach the user's goal. The information space is also important for inferring the information available for a specific user situation, for instance by determining colocated objects in the spatial context of the user's position. And finally, the information space is also connected to the system space. The system space covers technological factors, such as the telecommunication technology in use, network bandwidths and mobile device capabilities. These factors determine, in parts, the possibility, amount, and speed of information transmission to the mobile user. The mobile device in use and its properties constrain visualization-related factors, e. g., the number of displayable colors or data formats that can be rendered by the device. For further information on context modeling, see [6,7].

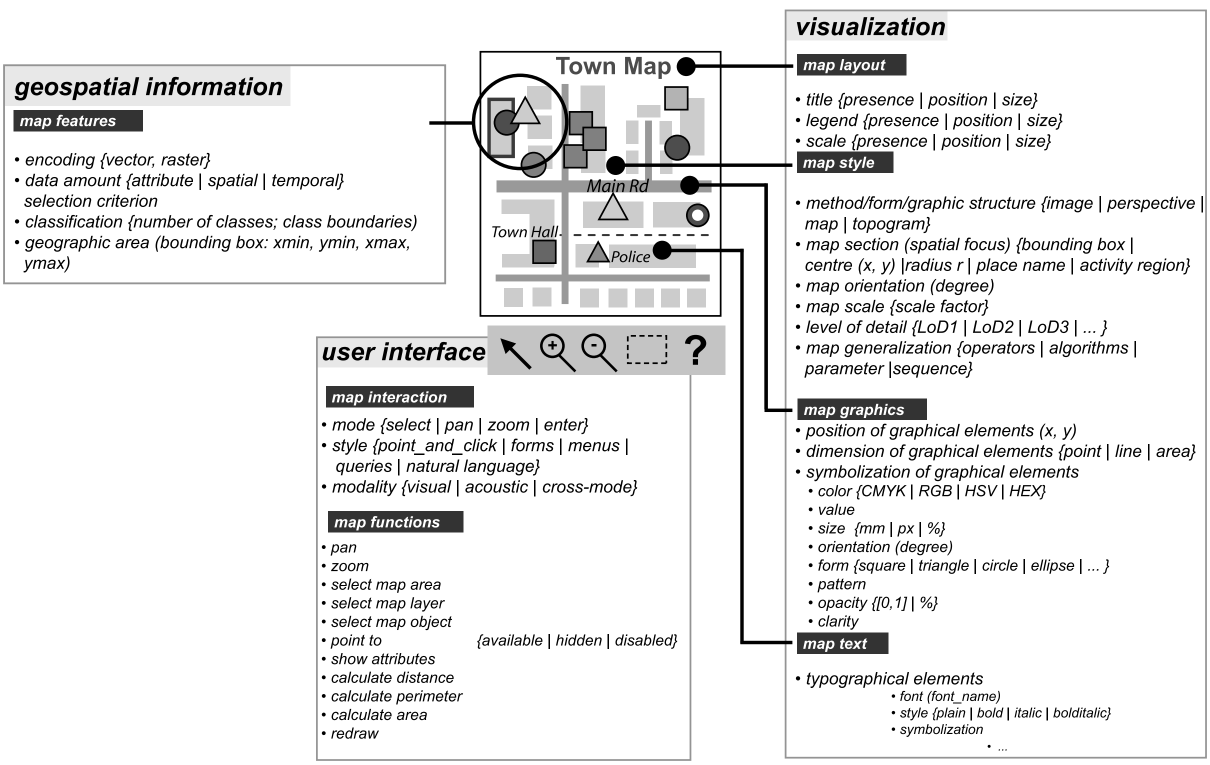
In mobile computing applications or services that have knowledge about the context they are used in or run are called context-aware. One way of capturing this knowledge about the context is the application of sensors. Sensors help a machine or software agent to sense parts of its environment and hence get information about this environment. In the case of the context of geographic information usage the most important and prominent sensor is a global positioning system (GPS) receiver. By determining the current position of the mobile device and thus of the mobile user it provides a special kind of context awareness, spatial awareness. Spatial awareness offers the opportunity for the simplest and most evident adaptivity of geographic information visualization for mobile usage. This kind of adaptivity is implemented in LBS where the information content is adapted to a specific location, in most cases a mobile user's current position. The simplest way of applying spatial awareness to the visualization of geographic information for a mobile user is a self-orienting map that moves the map section automatically based on the position received from a GPS receiver. This is implemented in car navigation systems where the map section is recentered when you are reaching the edge of the map.

Although the visualization of geographic information on mobile devices could be implemented in heavy, adaptable applications, it will rather be based on a service-oriented architecture (SOA). Such a web service concept for interoperable geospatial information services is specified by the Open Geospatial Consortium (OGC). The advantage of geospatial web services is their adaptability potential through the adjustment of service parameters.

The adaptation objects, i. e., the objects that can be changed for visualization either by the user or the system/service, to the adaptation target, i. e., the mobile usage context, are summarized in Fig. 4 using the example of a mobile map. Although visualization and its subcomponents are the central object of adaptation, the user interface and the geospatial information can be treated as separate adaptation objects of adaptive visualization.

The final building blocks of adaptive visualization are the adaptation methods that are responsible for adjusting one or several adaptation objects to the adaptation target.

The input to the adaptation method is one or more adaptation objects. The context parameter values are used to con-



Mobile Usage and Adaptive Visualization, Figure 4 Adaptable objects in adaptive visualization trol the adaptation method. The output of the adaptation method is a set of adapted objects. For the adaptation of the visualization of geographic information to the mobile usage context, the following adaptation methods are applicable:

- Selection method: this method selects map features depending on the usage context in order to reduce the map content and the information density. This method operates as a kind of filter. The filter criterion can be a spatial relation as in the case of LBS, or it could be a user preference stored in a user profile.
- Prioritization method: this method classifies the priority of selected information items with regard to the current usage context. The priority classes are based on the relevance of the information for different context factors, e. g., relevance for the current location or relevance for the current activity.
- Substitution method: this method substitutes one visualization form with an equivalent presentation form. A map could, for instance, be substituted by an image or abstract symbols might be replaced by pictorial symbols.
- Symbolization method: this method changes the symbolization of the visual elements by applying a different symbol style or by switching to a predefined design alternative.
- Configuration method: this method configures or reconfigures visual components in order to adjust the visualization to the usage context. For instance a different base map or a different scale might be selected depending on mobility-related factors such as means of transport or movement speed. The configuration method could also configure the user interface through hiding or aggregating functions or changing interaction modes.
- Encoding method: this method changes the encoding of the information (e. g., vector to raster) to be transmitted to a mobile device in dependence on technological factors such as the bandwidth available or the capabilities of the device.

  In the simplest case, a fixed rule base exists that triggers the adaptation methods and states what changes will be applied if specific context conditions are given. More advanced adaptive systems incorporate a learning component that dynamically changes the knowledge base in dependence on the system usage and performed user interactions.

Key Applications

Adaptive visualization, so far, is used mainly in the domain of mobile systems and services. The domains discussed here are mobile guide systems, LBS, and mobile geospatial web services.

Mobile Guide Systems

The adaptivity principle has been applied in many mobile guide systems [8,9]. The visualization in such guide systems is either adapted to the user characteristics or preferences (personalization) or to the usage context factors in general.

Location-Based Services

LBS are inherently adaptive by spatially filtering information for a specified location [10]. Different ways of adapting LBS are described in [11].

Mobile Geospatial Web Services

Most mobile geospatial web services include maps or other forms of visualization of geographic information to assist the user in way-finding, orientation or similar spatial tasks. Adaptive visualizations are employed in such services to improve their usability [7].

Future Directions

In principle, any visualization of geographic information can be adapted to the context of use. With the growing amount of geospatial data available the demand for the adaptation to specific needs and contexts will probably get stronger in the near future.

The adaptivity principle is especially important for the evolving mobile geospatial web services and their interfaces where usability is a crucial issue. A thoughtful implementation of adaptive behavior helps to reduce complexity and take some of the cognitive load inherent in mobile usage contexts. Adaptive visualization will play an important role in personalized geospatial web services and egocentric maps [12].

Cross References

► Visualizing Constraint Data
► Web Services, Geospatial

Recommended Reading

1. Reichenbacher, T.:, Mobile Cartography Adaptive Visualisation of Geographic Information on Mobile Devices. Verlag Dr. Hut, Munich (2004)
2. Browne, D., Totterdell, P., Norman, M. (eds.): Adaptive user interfaces. Academic Press, London (1990)
3. Oppermann, R. (ed.): Adaptive user support: ergonomic design of manually and automatically adaptable software. Lawrence Erlbaum, Hillsdale, NJ (1994)
4. Schneider-Hufschmidt, M., Kühme, T., Malinowski, U.(eds.): Adaptive user interfaces: principles and practice. North-Holland, Amsterdam (1993)

M

5. Brusilovsky, P.: Adaptive hypermedia. User Model. User-Adapt. Interact. **11**, 87–110 (2001)
6. Dey, A.K.: Understanding and using context. Personal Ubiquitous Comput. J. **5**, 4–7 (2001)
7. Sarjakoski, T., Nivala A.-M.: Adaptation to context — a way to improve the usability of mobile maps. In: Meng, L., Zipf, A., Reichenbacher, T. (eds.) Map-Based Mobile Services. Theories, Methods and Implementations, pp. 7–123. Springer, Heidelberg (2005)
8. Oppermann, R., Specht, M.: Adaptive mobile museum guide for information and learning on demand, in human-computer interaction. In: Bullinger, H.J., Ziegler, J. (eds.). Proceedings of HCI International '99, vol. 2: Communication, Cooperation, And Application Design, pp. 642–646. Erlbaum, Mahwah (1999)
9. Patalaviciute, V., et al.: Using SVG-based Maps for Mobile Guide Systems—A Case Study for the Design, Adaptation, and Usage of SVG-based Maps for Mobile Nature Guide Applications. (2005) Available via http://www.svgopen.org/2005/papers/MapsForMobileNatureGuideApplications/index.html
10. Krug, K., Mountain, D., Phan, D.: Location-based services for mobile users in protected areas. Geoinformatics, **6**(2), 26–29 (2003)
11. Steiniger, S., Neun, M., Edwardes, A.: Foundations of Location Based Services (2006). Available via http://www.geo.unizh.ch/publications/cartouche/lbs_lecturenotes_steinigeretal2006.pdf
12. Reichenbacher, T.: Adaptive egocentric maps for mobile users. In: Meng, L., Zipf, A., Reichenbacher, T. (eds.) Map-Based Mobile Services. Theories, Methods and Implementations, pp. 141–158. Springer, Heidelberg (2005)

MOBR

► Minimum Bounding Rectangle

Model Driven Architecture

► Modeling with Enriched Model Driven Architecture

Model Driven Development

► Modeling with Enriched Model Driven Architecture

Model Driven Engineering

► Modeling with Enriched Model Driven Architecture

Model Generalization

► Abstraction of GeoDatabases

Modeling and Multiple Perceptions

CHRISTINE PARENT[1], STEFANO SPACCAPIETRA[2], ESTEBAN ZIMÁNYI[3]
[1] University of Lausanne, Lausanne, Switzerland
[2] Swiss Federal Institute of Technology, Lausanne, Switzerland
[3] Free University of Brussels, Brussels, Belgium

Synonyms

Data modeling; Multiscale databases; Multirepresentation

Definition

Multirepresentation generalizes known concepts such as database views and geographic multiscale databases. This chapter describes the handling of multi-representation in the MADS (Modeling Application Data with Spatio-temporal features) data modeling approach. MADS builds on the concept of orthogonality to support multiple modeling dimensions. The structural basis of the MADS model is based on extended entity-relationship (ER) constructs. This is complemented with three other modeling dimensions: space, time, and representation. The latter allows the specification of multiple perceptions of the real world and modeling of the multiple representations of real-world elements that are needed to materialize these perceptions.

Historical Background

Traditional database design organizes the data of interest into a database schema, which describes objects and their relationships, as well as their attributes. At the conceptual level, the design task relies on well-known modeling approaches such as the ER model [1] and Unified Modeling Language (UML) [2]. These approaches only deal with classical alphanumeric data. The idea of using conceptual spatial and spatiotemporal data models emerged in the 1990s. Most proposals were extensions of either the ER (e. g., [3,4,5]), UML (e. g., [6,7]), or object-oriented data models (e. g., [8]). Spatiotemporal data models are the current focus of research and development, both in academia and in industry: They allow the representation of the past and future evolution of geographic objects as well as moving and deforming objects. Both features are essential for complex development issues such as environmental management and city planning.

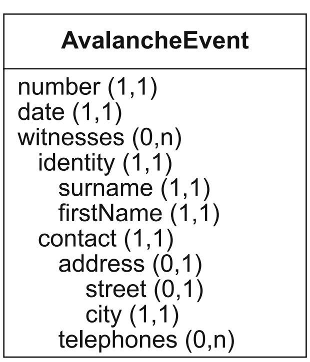
Most geographic applications confronted with a variety of user categories showing different requirements of the same data (e. g., different administrations involved in city management) also need another modeling dimension: multiple representations. Multiple representations allow, for example, the spatial feature of a city being described as a point and as an area, the former for use in statewide maps, the latter in local maps. Interest in supporting this functionality has emerged recently [5,9] and the concept of multi-representation is now popular in the research community and with user groups. The MADS proposal is playing an important role in establishing and advancing this trend.

Scientific Fundamentals

Definition 1: *conceptual data model.* A data model is conceptual if it enables a direct mapping between the perceived real world and its representation with the concepts of the model. In particular, a conceptual model does not have implementation-related concerns.

Definition 2: *data modeling dimension.* A data modeling dimension is a domain of representation of the real world that focuses on a specific class of phenomena. Examples of modeling dimensions include: data structure, space, time, and multirepresentation.

Definition 3: *orthogonality.* Modeling dimensions are said to be orthogonal if, when designing a database schema, choices in a given dimension do not depend on the choices in other dimensions. For example, it should be possible to record the location of a reservoir on a river (a spatial feature in the space dimension) irrespective of whether the reservoir, in the data structure dimension, has been modeled as an independent object or as an attribute of the river object. Orthogonality greatly simplifies the data model and its use, while enhancing its expressive power, i. e., its ability to represent all phenomena of interest. The orthogonality of multiple modeling dimensions is an essential characteristic of MADS. A detailed presentation of the model can be found in [5,10].



Modeling and Multiple Perceptions, Figure 1 A diagram of an object type with its attributes

Thematic Data Structure Modeling

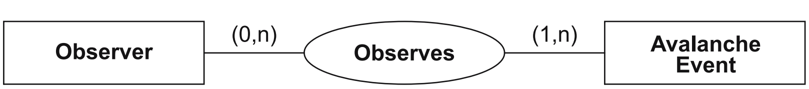
Definitions 4, 5, and 6: Object Types, Attributes, and Methods Database *objects* represent real-world entities of interest to applications. An *object type* defines the properties of interest for a set of objects that, from the application viewpoint, are considered as similar. Properties are either attributes or methods. An *attribute* is a property that is represented by a value in each object of the type. A *method* is a behavioral property common to all objects of the type.

Figure 1 illustrates an object type AvalancheEvent and its attributes. These may be *simple*, e. g., number for AvalancheEvent, or *complex* (i. e., composed of other attributes), e. g., witnesses.

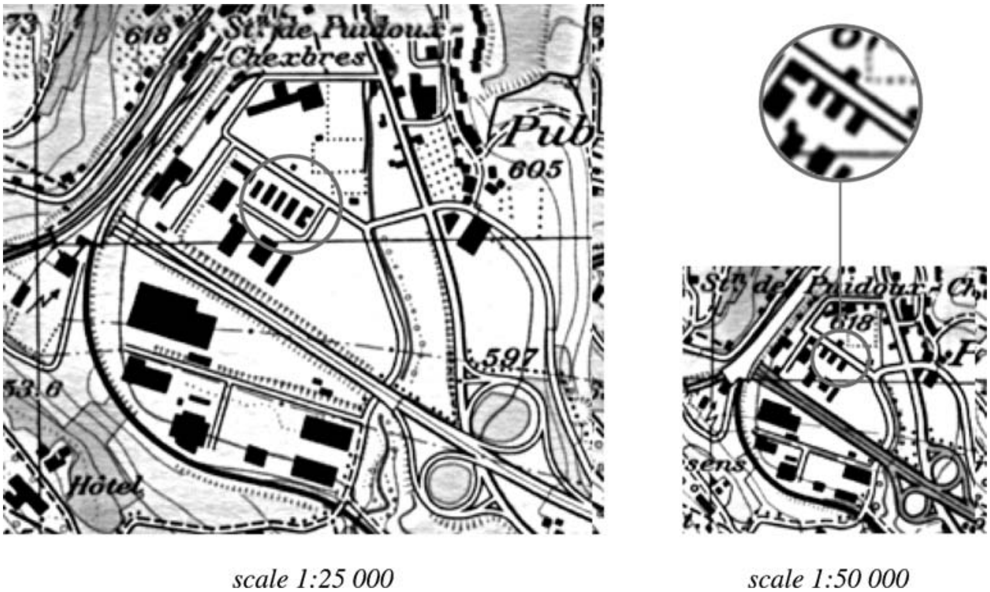
Definition 7: Cardinality *Attribute cardinality*, defined by two numbers (min, max), denotes the minimum and maximum number of values that an attribute may hold within an object. The maximum cardinality determines whether an attribute is *monovalued* or *multivalued*, i. e., whether it holds at most one or several values. The minimum cardinality determines whether an attribute is *optional* (min = 0) or *mandatory* (min > 0), i. e., whether an object may hold no value or must hold at least one value.

Definitions 8 and 9: Relationship Types and Roles Relationships represent real-world links between objects that are of interest to the application. Relationships that, from the application perspective, have the same characteristics are grouped in relationship types. Roles represent the involvement of an object type into a relationship type. A relationship type defines two or more roles: They are *n*-ary, *n* being the number of roles. Figure 2 shows an example of a binary relationship of the type Observes.

Cardinality constraints define the minimum and maximum number of relationships that may link an object in a role. For example, in Fig. 2 the (0,*n*) cardinalities on the role associated to the Observer object type express that an observer may have never observed an avalanche, and that an observer may have observed any number of avalanches. As object types, relationship types may be described by properties (attributes and methods). The generic term *instance* denotes either an object or a relationship: an object (relationship) is an instance of the object (relationship) type it belongs to. MADS identifies two basic kinds of relationship types, association and multiassociation, described next.



Modeling and Multiple Perceptions, Figure 2 A diagram showing a relationship type linking two object types



Modeling and Multiple Perceptions, Figure 3 An example situation calling for a multiassociation relationship

Modeling and Multiple Perceptions, Figure 4 An example of a multiassociation type

Definition 10: Association Types An *association type* is a relationship type such that each role links exactly one instance of the linked object type.

Associations are the usual kind of relationships. However, in some situations the association relationship does not allow accurate representation of real-world links existing between objects. Figure 3 shows two maps of the same area at different scales. Focusing on the area within the superimposed circles, the left-hand, more detailed, map shows five aligned buildings whereas at the same location the right-hand map shows only three. Suppose that the application stores the five buildings in the left-hand map as instances of the BuildingScale15'000 object type, and the three buildings in the right-hand map as instances of the BuildingScale25'000 object type. If the application requires the correlation of cartographic representations, at different scales, of the same real-world entities, some link must relate the five instances of BuildingScale15'000 to the three instances of BuildingScale25'000. The multiassociation construct allows directly representation of such a link.

Definition 11: Multiassociation Types A *multiassociation type* is a relationship type such that each role links a nonempty collection of instances of the linked object type.

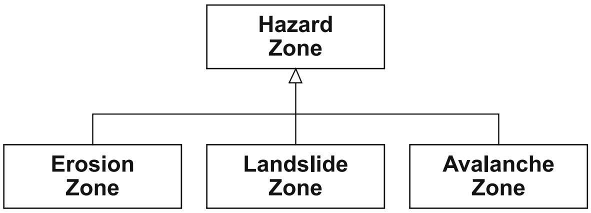
Consequently, each role in a multiassociation type bears two pairs of (min, max) cardinalities. A first pair is the conventional one, defining for each object instance how many relationship instances it can be linked to via the role. The second pair defines for each relationship instance how many object instances it can link with this role. Its value for minimum is at least 1. Using a multiassociation type, the above correspondence between cartographic buildings can be modeled as shown in Fig. 4.

Apart from the additional cardinality constraints, multiassociation types share the same features as association types.

Definition 12: is-a Links The *is-a* (or generalization/specialization) link relates two object or two relationship types, a generic one (the *supertype*) and a specific one (the *subtype*). It states that the subtype describes a subset of the real-world instances described by the supertype, and this description is a more precise one.

Figure 5 shows a generalization hierarchy of object types with three is-a links. The generic object type representing hazard zones is specialized in subtypes representing landslide, erosion, and avalanche zones.

A well-known characteristic of is-a links is *property inheritance*: All properties and links defined for the supertype also hold for the subtype. An immediate benefit of inher-



Modeling and Multiple Perceptions, Figure 5 Object types connected by is-a links

itance is *type substitutability*, i. e., enforcing the fact that wherever an instance of a type can be used in some data manipulation, an instance of any of its subtypes can be used instead.

Describing Space and Time Using the Discrete View

In MADS, space and time description is orthogonal to data structure description, which means that the description of a phenomenon may be enhanced by spatial and temporal features whatever data structure (i. e., object, relationship, attribute) has been chosen to represent it.

Definition 13: Discrete View The *discrete* (or *object*) *view* of space and time defines the spatial and temporal extents of the phenomena of interest. The *spatial extent* is the set of points that the phenomenon occupies in space, while the *temporal extent* is the set of instants that it occupies in time.

Specific data types support the manipulation of spatial and temporal values, like a point, a surface, or a time instant. MADS supports a hierarchy of spatial data types and another hierarchy of temporal data types. Generic data types allow the description of object types whose instances may have different types of spatial or temporal extents. For example, a River object type may contain large rivers with an extent of type Surface and small rivers with an extent of type Line. Examples of spatial data types are: Geo (), the most generic spatial data type, Surface (), and SurfaceBag (). The latter is useful for describing objects with a nonconnected surface, like an archipelago. Examples of temporal data types are: Instant (), TimeInterval (), and IntervalBag (). The latter is useful for describing the periods of activity of noncontinuous phenomena.

In MADS, temporality associated to object/relationship types or to attributes corresponds to *valid time*, which conveys information on when a given fact of the database is considered valid from the application viewpoint.

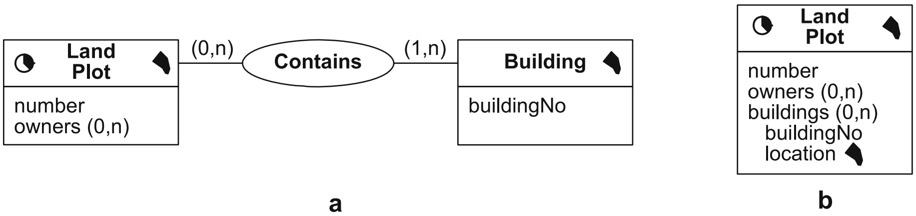
Definition 14: Spatial, Temporal, Spatiotemporal Objects and Relationships A *spatial* (and/or *temporal*) *object type* is an object type that holds spatial (and/or temporal) information pertaining to the object as a whole.

For example, in Fig. 6a both object types are spatial as shown by the Surface () icon, while only LandPlot is temporal as shown by the TimeInterval () icon. Following common practice, an object type is called *spatiotemporal* if either it has both a spatial and a temporal extent, separately, or has a time-varying spatial extent (i. e., its spatial extent changes over time and the history of extent values is recorded).

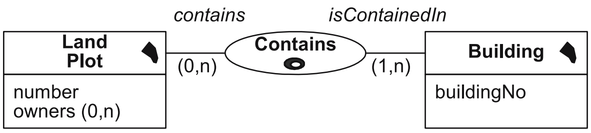
Similarly, *spatial*, *temporal*, and *spatiotemporal relationship types* hold spatial and/or temporal information pertaining to the relationship as a whole, as for an object type. For example, in Fig. 2, the Observes relationship type can be defined as temporal, of the type Instant, to record when observations are made.

Spatial and temporal information at the object- or relationship-type level is kept in dedicated system-defined attributes: *geometry* for space and *lifecycle* for time. Geometry is a spatial attribute (see below) with any spatial data type as domain. When representing a moving object, geometry is a time-varying spatial attribute. On the other hand, the lifecycle allows users to record when an object (or link) was (or is planned to be) created and deleted. It may also support recording that an object is temporarily suspended, like an employee who is on temporary leave. Therefore the lifecycle of an instance says for each instant what is the status of the corresponding real-world object (or link) at this instant: scheduled (its creation is planned later), active, suspended (it is temporarily inactive), disabled (definitively inactive).

Definition 15: Spatial, Temporal, Spatiotemporal Attributes A *spatial* (*temporal*) *attribute* is a simple attribute whose domain of values is one of the spatial (temporal) data types. A *spatiotemporal attribute* is a time-



Modeling and Multiple Perceptions, Figure 6 **a**, **b** Alternative schemas for land plots and buildings



Modeling and Multiple Perceptions, Figure 7 A topological relationship of the type **Inclusion** (⬬)

varying spatial attribute, i. e., a spatial attribute whose value changes over time and the history of its values is recorded (see Definition 17).

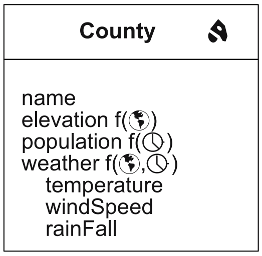
Each object and relationship type, whether spatiotemporal or not, may have spatial, temporal, and spatiotemporal attributes. For example, in Fig. 6b the **LandPlot** object type includes, in addition to its spatial extent, a complex and multivalued attribute **buildings** whose second component attribute, **location**, is a spatial attribute describing, for each building, its spatial extent.

Constraining Relationships with Space and Time Predicates

The links among spatial (temporal) object types often describe a spatial (temporal) constraint on the spatial (temporal) extents of the linked objects. For example, designers may want to enforce each **Contains** relationship of Fig. 6a to link a pair of objects provided that the spatial extent of the land plot contains the spatial extent of the building. In MADS, this can be done by defining **Contains** as a constraining relationship of the type topological inclusion, as shown in Fig. 7 by the ⬬ icon.

Definition 16: Constraining Relationships *Constraining relationships* are binary relationships enforcing the geometries or lifecycles of the linked objects types to comply with a topological or synchronization constraint.

Figure 8 shows a temporal synchronization relationship of the type **During**, stating that observers can observe avalanche events only while they are on duty. Relationship types may simultaneously bear multiple constraining semantics. For example, the **Intersects** relationship type shown in Fig. 9 enforces both a topological overlapping and a synchronization overlapping constraint.



Modeling and Multiple Perceptions, Figure 10 An object type with varying attributes

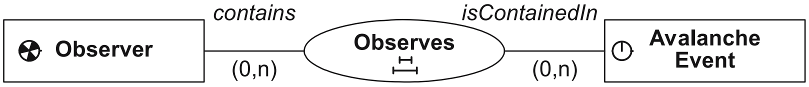
Describing Space and Time Using the Continuous View

Beyond the discrete view, there is a need to support another perception of space and time, the *continuous* (or *field*) *view*.

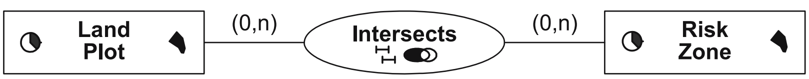
Definition 17: Continuous View, Varying Attribute In the *continuous view*, a phenomenon is perceived as a function associating to each point (instant) of a spatial (temporal) extent a value. In the MADS model the continuous view is supported by space (and/or time) *varying attributes*, which are attributes whose value is a function. The domain of the function is a spatial (and/or temporal) extent. Its range can be a set of simple values (e. g., **Real** for temperature, **Integer** for rainfall, **Point** for a moving car), or a set of composite values if the attribute is complex as, in Fig. 10, **weather**. If the attribute is multivalued, the range is a powerset of values.

Figure 10 shows examples of varying attributes and their visual notation in MADS. For instance, **elevation** is a space-varying attribute defined over the geometry of the county. It provides for each geographic point of the county its elevation. An example of a time-varying attribute is **population**, which is defined over a time interval, e. g., [1900, 2008]. Then, **weather** is a space and time-varying attribute which gives for each point of the spatial extent of the county and each instant of a time interval a composite value describing the weather at this location and this instant. Attributes that are space and time-varying are also called *spatiotemporal attributes*.

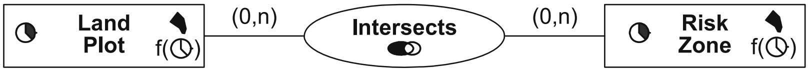
A constraining topological relationship may link moving or deforming objects, i. e., spatial objects whose geome-



Modeling and Multiple Perceptions, Figure 8 A synchronization relationship type of the **During** kind



Modeling and Multiple Perceptions, Figure 9 A topological **Overlap** and synchronization **Overlap** relationship type



Modeling and Multiple Perceptions, Figure 11 An example of a topological relationship that links spatial object types with deforming geometries

tries are time-varying. Figure 11 shows an example. In this case two possible interpretations can be given to the topological predicate, depending on whether it must be satisfied either in at least one instant or in every instant belonging to both time extents of the varying geometries [11]. Applied to the example of Fig. 11, the two interpretations result in accepting in the relationship **Intersects** only instances that link a land plot and a risk zone such that their geometries intersect for at least one instant or for every instant belonging to both life spans. When defining the relationship type, the designer has to specify which interpretation holds.

Supporting Multiple Perceptions and Multiple Representations

Databases store representations of real-world phenomena that are of interest to a given set of applications. However, while the real world is supposed to be unique, its representation depends on the intended purpose.

Definition 18: Perceptions and Representations Each application has a peculiar *perception* of the real world of interest. These perceptions may vary both in terms of what information is to be kept and in terms of how the information is to be represented. Fully coping with such diversity entails that any database element may have several descriptions, or *representations*, each one associated to the perceptions it belongs to. Both metadata (descriptions of objects, relationships, attributes, is-a links) and data (instances and attribute values) may have multiple representations. There is a bidirectional mapping between the set of perceptions and the set of representations. This mapping links each perception to the representations perceived through this perception.

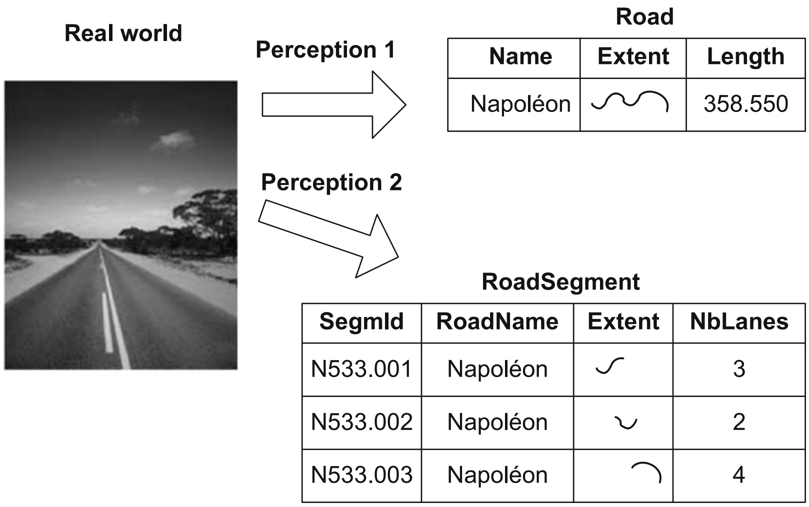
Classic databases do not support the mapping between perceptions and representations. They usually store for each real-world entity or link a unique, generic representation, hosting whatever is needed to globally comply with all application perceptions. An exception exists for databases supporting generalization hierarchies, which allow the storage of several representations of the same entity in increasing levels of specificity. These classic databases have no knowledge of perceptions. Hence the system cannot provide any service related to perception-dependent data management. Applications have to resort to the view mechanism to define and extract data sets that correspond to their own perception of the database. But still, each view is a relational table or object class. They cannot make up a whole perception, which is a subset of the database containing several object types related by relationship types and is-a links. Instead, MADS explicitly supports multiple perceptions for the same database.

Definition 19: Multiperception Databases A *multiperception database* is a database where designers and users have the ability to define and manipulate several perceptions of the database. A multiperception database stores one or several representations for each database element, and records for each perception the representations it is made up.

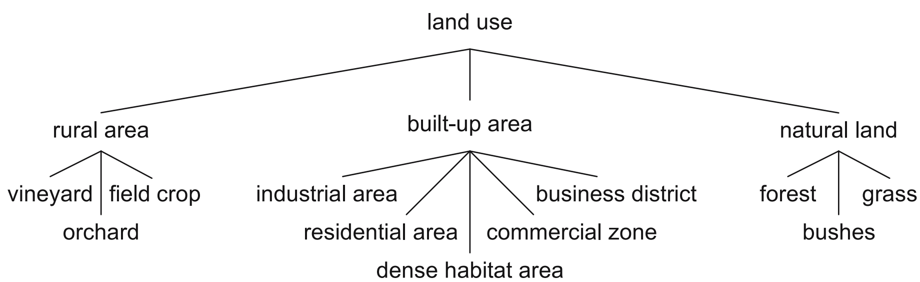
Geographical applications have strong requirements in terms of multiple representations. For example, cartographic applications need to keep multiple geometries for each object, each geometry corresponding to a representation of the extent of the object at a given *spatial resolution*. The resolution of a spatial database is the minimum size of any spatial extent stored in the database. Resolution is closely related to the scale of the maps that are produced from the database. The scale of a printed map is the amount of reduction between the real world and its graphic representation in the map. Multiscale representations are needed as there is still no complete set of algorithms for *cartographic generalization* [12], i. e., the process to automatically derive a representation at a less-detailed resolution from a representation at a more precise resolution.

Geographical databases are also subject to classical semantic resolution differences. For example, an application may see a road as a single object, while another one may see it in more detail as a sequence of road sections, each one represented as an object, as in Fig. 12. As another example, geographical databases need to support *hierarchical value domains* for attributes, where values are chosen depending on the level of detail. In the hierarchical domain for **land use**, Fig. 13, **orchard** and **rural area** are two representations of the same value at different resolutions.

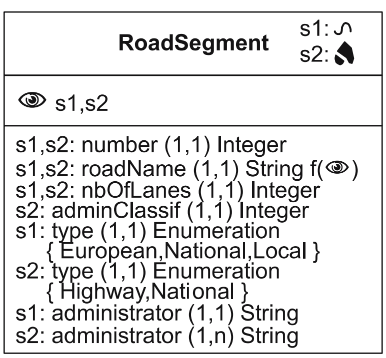
In MADS, each perception has a user-defined identifier, called its *perception stamp*, or just *stamp*. In the sequel, perception stamps are denoted as s1, s2, …, sn. From data definitions (metadata) to data values, anything in a database (object type, relationship type, attribute, role, instance, value) belongs to one or several perceptions. Stamping an element of the schema or of the database defines for which perceptions the element is relevant. In



Modeling and Multiple Perceptions, Figure 12 Two different perceptions of the same reality, leading to different representations



Modeling and Multiple Perceptions, Figure 13 An example of a hierarchical domain



Modeling and Multiple Perceptions, Figure 14 An illustration of a birepresentation type, defined for perceptions s1 and s2

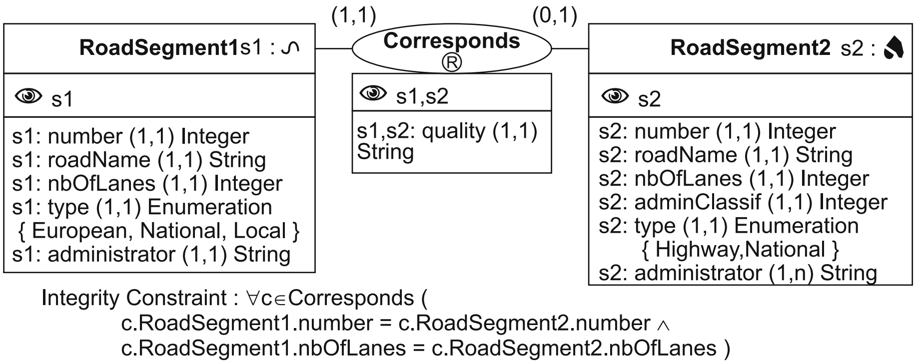
the diagrams, e. g., Fig. 14, the box identified by the 👁 icon defines the set of perceptions for which this type is valid. Similarly, the specification of the relevant stamps is attached to each attribute and method definition.

There are two complementary techniques to organize multiple representations. One solution is to build a single object type containing several representations, the knowledge of "which representation belongs to which perception" being provided by the stamps of the properties of the type. Following this approach, in Fig. 14 the designer has defined a single object type RoadSegment, grouping two representations, one for perception s1 and one for perception s2.

Definition 20: Multirepresentation Object/Relationship Types An object or relationship type is *multirepresentation* if at least one of its characteristics has at least two different representations. The characteristic may be at the schema level (e. g., an attribute with different definitions) or at the instance level (i. e., different sets of instances or an instance with two different values).

The alternative solution to organize multiple representations is to define two separate object types, each one bearing the corresponding stamp(s) (cf. Fig. 15). The knowledge that the two representations describe the same entities is then conveyed by linking the object types with a relationship type that holds an *interrepresentation* semantics (indicated by the Ⓡ icon). In the example of Fig. 15, the same real-world road segment is materialized in the database as two object instances, one in RoadSegment1 and one in RoadSegment2. Instances of the relationship type Corresponds tell which object instances represent the same road segment.

The actual representation of instances of multirepresentation object types changes from one perception to another.



Modeling and Multiple Perceptions, Figure 15 The **RoadSegment** type (from Fig. 14) split into two monorepresentation object types and an interrepresentation relationship type

Consider the object type **RoadSegment** of Fig. 14. The spatial extent of the type is represented either as a surface (the more precise description, perception s2) or as a line (the less precise description, perception s1) depending on resolution. Furthermore, perception s1 needs the attributes **number**, **roadName**, **numberOfLanes**, **type**, and **administrator** (denoting the maintenance firm in charge). Perception s2 needs the attributes **number**, **roadName**, **numberOfLanes**, **adminClassification**, **type**, and **administrator**. While the road segment number and the number of lanes are the same for s1 and s2, the name of the road is different, although a string in both cases. For instance, the same road may have name "RN85" in perception s1 and name "Route Napoléon" in s2. We call this a *perception-varying attribute* (see below), identified as such by the **f**(👁) notation. The **type** attribute takes its values from predefined sets of values, the sets being different for s1 and s2. Several administrators for a road segment may be recorded for s2, while s1 records only one.

Definition 21: Perception-Varying Attribute An attribute is perception-varying if its value in an instance may change from one perception to another. A perception-varying attribute is a function whose domain is the set of perceptions of the object (or relationship) type and whose range is the value domain defined for this attribute. These attributes are the counterpart of space-varying and time-varying attributes in the space and time modeling dimensions.

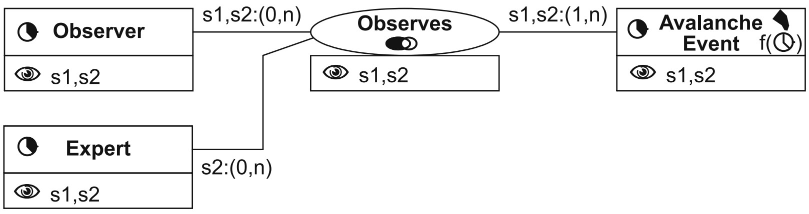
Stamps may also be specified at the instance level. This allows the defining of different subsets of instances that are visible for different perceptions. For example, as the object type **RoadSegment** in Fig. 14 has two stamps, it is possible to define instances that are only visible to s1, instances that are only visible to s2, and instances that are visible to both s1 and s2.

Relationship types can be in multirepresentation, like object types. Their structure (roles and association/multiassociation kind) and semantics (e. g., topology, synchronization) may also have different definitions depending on the perception. Figure 16 shows an example of different semantics, where the designer defined the relationship **GivesAccess** as (1) a topological adjacent relationship type for perception s1, and (2) a plain relationship without any peculiar semantics or constraint for perception s2.

A relationship type may have different roles for different perceptions. For example, Fig. 17 shows that an observation is perceived in s1 as a binary relationship between an observer and an avalanche event, while perception s2 sees the same observation as a ternary relationship, also involving the expert who has validated the observation.

M

Modeling and Multiple Perceptions, Figure 16 A relationship type with two different semantics: topological adjacency for s1 and plain relationship for s2



Modeling and Multiple Perceptions, Figure 17 A relationship type with a role specific to perception s2

Key Applications

Multirepresentation databases are an essential feature for all applications that need to provide different categories of users with different sets of data, organized in different ways, without imposing a single centralized model of the real world for the whole enterprise. Multirepresentation databases play a key role in guaranteeing and maintaining the consistency of a database with decentralized control and autonomy of updates from different user categories. Multiscale map production applications are one example of a domain where this feature is highly desirable. Management and analysis of municipal, regional, and statewide databases are other examples where the same geographic area is input to a variety of uses. Domain-oriented (e. g., environmental) interorganizational as well as international applications call for integrated databases that provide autonomous usage by the contributing organizations and countries.

Future Directions

Considering the large number of perceptions that some huge databases may have to support, an investigation into how perceptions may be organized and structured (as objects of interest per se) is planned.

Another direction for future research is adding complementary modeling dimensions to the MADS model, such as the multimedia dimension, the precision dimension (supporting, for example, fuzzy, imprecise, and approximate spatial and temporal features), and the trust or data quality dimension.

Finally, spatiotemporal and trajectory data warehousing represent wide-open research domains to develop efficient support for geographically based decision making.

Cross References

► Modeling with ISO 191xx Standards

Recommended Reading

1. Chen, P.P.: The entity-relationship model: towards a unified view of data. ACM Trans. Database Syst. **1**, 9–36 (1976)
2. Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language, Reference Manual, 2nd edn. Addison-Wesley, Boston (2005)
3. Bédard, Y., Pageau, J., Caron, C.: Spatial data modeling: the Modul-R formalism and CASE technology. In: Proceedings of the ISPRS Symposium, Washington, 1–14 Aug 1992
4. Tryfona, N., Jensen, C.S.: Conceptual Data Modeling for Spatiotemporal Application. GeoInformatica **3**(3), 245-268 (1999)
5. Parent, C., Spaccapietra, S., Zimányi, E.: Conceptual Modeling for Traditional and Spatio-temporal Applications: The MADS Approach. Springer, Berlin, Heidelberg (2006)
6. Price, R., Ramamohanarao, K., Srinivasan, B.: Spatio-temporal Extensions to Unified Modeling Language. In: Proceedings of the Workshop on Spatio-temporal Data Models and Languages, IEEE DEXA'99 Workshop, Florence, Italy, 1–3 Sept 1999
7. Bédard, Y., Larrivée, S., Proulx, M.J., Nadeau, M.: Modeling geospatial databases with plug-ins for visual languages: a pragmatic approach and the impacts of 16 years of research and experimentations on perceptory. In: Wang, S. et al. (eds.) ER Workshops, Shanghai, China. Lecture Notes in Computer Science, vol. 3289, pp. 17–30. Springer, Berlin, Heidelberg (2004)
8. David, B., Raynal, L., Schorter, G.: GeO2: Why objects in a geographical DBMS? In: Abel, D.J., Ooi, B.C. (eds.) Proceedings of the 3rd International Symposium on Advances in Spatial Databases, SSD'93, Singaporer, 23–25 June 1993. Lecture Notes in Computer Science, vol. 692, pp. 264–276. Springer Berlin, Heidelberg (1993)
9. Bédard, Y., Bernier, É.: Supporting multiple representations with spatial databases view management and the concept of VUEL. In: The Joint Workshop on Multi-Scale Representation of Spatial Data, ISPRS WG IV/3, ICA Commission. On Map Generalization, Ottawa, Canada, 7–8 July 2002
10. Parent, C., Spaccapietra, S., Zimányi, E.: The MurMur Project: modeling and querying multi-represented spatio-temporal databases. Inf. Syst. **31**, 733–769 (2006)
11. Erwig, M., Schneider, M.: Spatio-temporal predicates. IEEE Trans. Knowledge Data Eng. **14**, 881–901 (2002)
12. Weibel, R., Dutton, G.: Generalizing spatial data and dealing with multiple representations. In: Longley, P.A., Goodchild, M.F., Maguire, D.J., Rhind, D.W. (eds.) Geographical Information Systems: Principles, Techniques, Management and Applications, 2nd edn. vol. 1, pp. 125–155. Wiley, New York (1999)
14. Koubarakis, M. et al. (eds.): Spatio-temporal Databases: The Chorochronos Approach. Lecture Notes in Computer Science, vol. 2520. Springer, Berlin, Heidelberg (2003)
15. Güting, R.H., Schneider, M.: Moving Objects Databases. Morgan Kaufmann, Amsterdam (2005)
16. Malinowski, E., Zimányi, E.: Advanced data warehouse design: From conventional to spatial and temporal applications. Springer, Berlin, Heidelberg (2007, in press)

Modeling Cycles in Geospatial Domains

JORGE CAMPOS
Department of Exact Sciences,
Salvador University – UNIFACS, Salvador, Brazil

Synonyms

Event, cyclic; Event, recurrent; Event, periodic

Definition

In a geospatial context, cycles can be defined as regularly repeated phenomena or events. According to the distribution of this kind of happening over time, cycles can be classified as *strong periodic*, *near periodic*, or *intermittent*. Strong periodicity refers to cycles in which the duration of every occurrence of the event is the same as is the temporal interval between two consecutive occurrences. Train schedules and tidal movements are examples of events with a strong periodic pattern of repetition. The second class of periodicity deals with events that occur regularly, but the occurrences do not necessarily have the same duration nor are they equally spaced in time. For this kind of cycle, the duration of events and the temporal gap between each of them may vary in a stochastic or deterministic manner. The spill of a geyser is an example of a near periodic event. The third class of periodicity deals with events that occur in a more irregular fashion or not regularly enough to be predicted with any degree of certainty. The occurrences of category 5 hurricanes over the Gulf of Mexico are an example of an intermittent phenomenon.

Main Text

Currently, there are many kinds of specialists dealing with or studying phenomena that repeat regularly. Engineers and social scientists, for example, cooperate to synchronize the schedules of the public transportation system with people's daily routine in urban environments. Environmental, social and financial experts struggle to predict the effect of global warming on seasonal precipitations and, therefore, on the occurrences of floods. Biologists and life scientists are working to identify relations between birds' migration cycles and the recurrent occurrences of some human and animals' diseases. Thus, there are many research questions driving the efforts of a wide range of specialists who deal with cyclic phenomena. In the database area, for example, there are needs for conceptual models and formalisms for expressing periodicity and for constructing queries about periodic data. In knowledge representation and temporal reasoning, there are concerns about periodic-based temporal constraints and modeling cyclic temporal relations [1], which are important subjects for scheduling and for constraint satisfaction problems involving cyclic events [2]. In the knowledge discovery and data mining field, there are requests for the support of mining temporal data to discover trends, patterns, and relationships between cyclic recurrent events [3].

The geographic information science research community has had a strong interest in capturing dynamic or time-varying phenomena in geographic space and representing such phenomena in spatio-temporal data models. Most successful models have adopted either spatio-temporal extensions of the entity-relationship or object-oriented models, or an event- and process-based approach. Whichever model is used to represent the temporal dimension of the phenomena, few studies have focused on happenings that repeat themselves in a cyclic manner.

Cross References

► Geographic Dynamics, Visualization And Modeling
► Processes and Events
► Time Geography

Recommended Reading

1. Hornsby, K., Egenhofer, M., Hayes, P.: Modeling cyclic change. In: Advances in Conceptual Modeling, Proceedings of the ER'99 Workshops, Paris, France., Lecture Notes in Computer Science, vol. 1727, pp. 98–109. Springer-Verlag, Berlin (1999)
2. Isli, A.: On deciding consistency for CSPs of cyclic time intervals. In: Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference, Key West, FL, 21–23 May 2001
3. Roddick, J.F., Spiliopoulou M.: A bibliography of temporal, spatial and spatio-temporal data mining research. SIGKDD Explor. Newsl. **1**, 34–38 (1999)

Modeling Geospatial Application Database

► Modeling with ISO 191xx Standards

Modeling Geospatial Databases

► Modeling with ISO 191xx Standards

Modeling with a UML Profile

JUGURTA LISBOA FILHO[1], CIRANO IOCHPE[2]
[1] Department of Computer Science/Information Systems, Federal University of Vicosa (UFV), Vicosa, Brazil
[2] Institute of Computer Science/Information Systems, Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil

Synonyms

Geographic database conceptual modeling; Unified Modeling Language-Geoframe Modeling Language

Definition

A spatial database management system (SDBMS) provides storage structures and basic operations for spatial data manipulation, whereas geographic information systems (GIS) provide the mechanisms for analysis and visualization of geographic data [1]. In this way, geographic databases (GeoDB) are collections of georeferenced spatial data, stored by SDBMS and manipulated by GIS.

GeoDB, as any database, must be designed following the traditional database design methodology that includes the conceptual, logical and physical design phases [2]. To draw up a data schema during the conceptual phase, a conceptual modeling language must be used. A strong tendency exists in computer science to adopt the *Unified Modeling Language* (UML) [3] as a system modeling standard based on the object-oriented paradigm, and more specifically the UML class diagram for database design. However, for GeoBD design, it is necessary to extend UML with new elements that enable the modeling of spatial–temporal characteristics of geographical phenomena. UML is a naturally extensible language, in other words, it has its own constructs allowing its extension. The stereotype concept, one of the *UML extension mechanisms*, allows the definition of new specific model elements generating a profile tailored for a particular problem domain [3]. There are some UML extensions for GeoDB modeling [4,5,6]. To exemplify a spatial UML profile, described here is the Spatialtemporal UML-GeoFrame modeling language, which extends the UML, generating a profile of stereotypes to support the GeoDB conceptual modeling.

Historical Background

GIS were originated outside the computer science field, unlike most software technologies developed in the last decades such as the operating systems based on windows, DBMS, fourth generation languages, CAD, CAM and CASE tools, Office Automation Systems (OIS), and more recently the World Wide Web (WWW) with the software revolution due to the explosion of internet use.

One of the consequences of this historical origin is that most GIS application designers are their own users, who have the evolutionary approach as their main software developmental methodology, and whose main focus of attention is geospatial data acquisition and analysis. Thus, the old raster-vector debate [7] prevailed for a long time as an important theme in GIS conferences. Consequently, methodologies developed in the software engineering field are frequently not used in GIS application design, causing great losses in the quality of the produced systems and high maintenance costs.

An alternative to reduce these problems is the use of a database design methodology. Consequently, during the 1990s, several extensions of specific conceptual modeling languages for GIS applications were proposed in the literature. Initially these modeling languages were based on the entity-relationship model (ER), proposed by Peter Chen [8] or one of its extensions (e. g., Merise and Enhanced Entity-Relationship, EER). A few modeling languages were based on the semantic data model IFO [9]. At that time, the use of the object-oriented paradigm in system development was becoming more and more popular. Accordingly, several authors used as their base object-oriented design methods, such as OMT [10] and OOA [11], proposing extensions for the modeling of spatiotemporal aspects of geographical phenomena [4,5,6].

With the aim of standardizing the different existent graphic notations and defining a basic group of model constructs for software systems, in 1996 three great experts on object-oriented modeling joined their approaches to create the UML [12]. Consequently, by 1999 some UML extensions to GeoDB modeling came out, some of them supported by CASE tools (e. g., Perceptory [4], ArgoCASEGEO [13]).

The UML-GeoFrame modeling language [6] is described here, to exemplify a spatial UML profile, and show how UML can be naturally extended by its own extension mechanism, named stereotype. Unlike other conceptual modeling languages that seek constructs' completeness, so as to consider almost all modeling possibilities of geographical phenomena in different dimensions (descriptive, spatial and temporal), the UML-GeoFrame has as its inspiration the simplicity of the ER model and proposes the smallest possible group of stereotypes to assist the main requirements of GeoDB modeling, but at the same time allowing understanding by nonspecialized users, through a quite simple and instinctive graphic notation.

Scientific Fundamentals

UML-GeoFrame: a Modeling Language for Geographic Databases

A conceptual data modeling language provides a formal base (notational and semantics) for tools and techniques used in data modeling. Data modeling is the abstraction process where only the essential elements of the observed reality are emphasized, the nonessential elements being discarded. The process of conceptual database modeling comprises the description of the possible data content, besides structures and constraints applicable to them. This database description is based on the semantic constructs provided by a conceptual data modeling language.

The UML-GeoFrame, originally presented in [6], is based on a hierarchical class structure that makes up the conceptual GeoFrame framework. The GeoFrame provides the fundamental elements present in any GeoDB, whereas the UML class diagram provides the semantic constructs for a conceptual modeling language. This integration enables GeoDB design in a graphic language easily understandable by the users.

The result of the modeling process is a conceptual data schema that expresses "what" will be stored in the database and not "how" the data will be stored. A conceptual data schema becomes therefore an abstraction of the real world that is being modeled (miniworld). Consequently, every element of the reality to be modeled in the conceptual data schema must be stored in the GeoDB. In the same way, every object stored in a GeoDB must have been represented in the conceptual data schema, but this does not often happen.

A GeoDB stores three large data categories: conventional data without geographic reference (e. g., a property owner), geographic phenomena perceived in object view (e. g., cities, roads, parcels), and geographic phenomena perceived in field view (e. g., temperature, soil type, relief). The main UML-GeoFrame's contribution consists of providing a construct group that enables the designer to carry out the modeling of the geographic phenomena perceived in field view suitably. The geographic phenomena perceived in the object view and also the conventional objects are modeled in the same way as most existent modeling languages.

Therefore, the UML-GeoFrame uses the same constructs of the UML class diagram, such as classes and subclasses containing attributes and operations, and associations between classes, also enabling the specification of aggregations and compositions [12].

The GeoFrame Framework

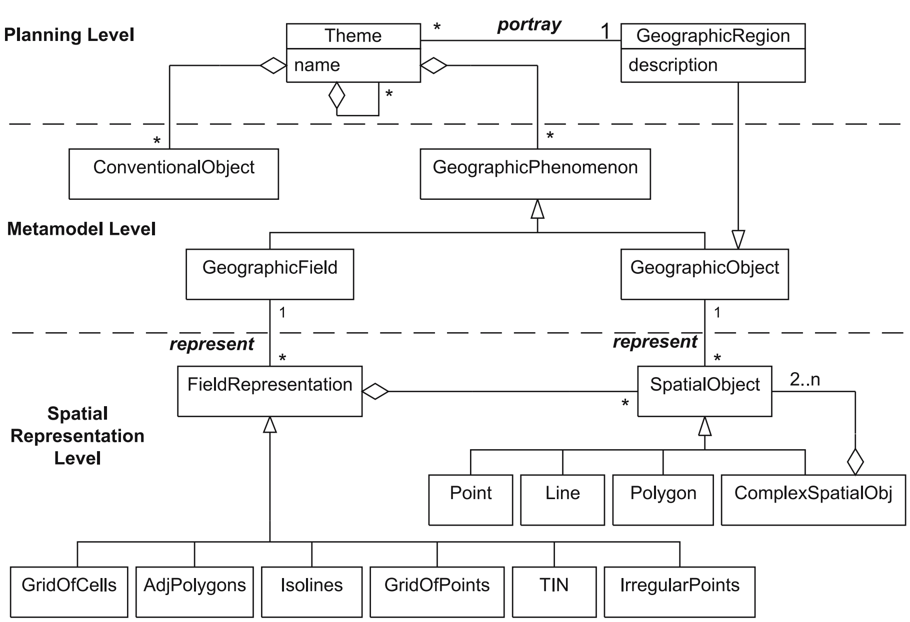
GeoFrame (Fig. 1) is a conceptual framework that provides a basic class diagram to assist the designer on the first steps of the conceptual data modeling of a new GIS application. The mutual use of a UML class diagram and GeoFrame allows the solution of most requirements of GIS application modeling.

The GeoFrame class diagram has three abstraction levels. The first is the planning level, which comprises the GeographicRegion class, whose instances correspond to the application interest areas, and the Theme class, describing the several themes that will portray this area. The metamodel level comprises the most generic classes of the geographic reality, which are divided in two categories, the conventional objects (without spatial representation) and the geographic phenomena, that comprise the geographic phenomena perceived in field view and the geographic phenomena perceived in object view. The third level includes the classes of objects that enable the designer to abstract the type of spatial representation that will be specified for each type of geographic phenomenon, multiple representations being possible.

To exemplify both the UML-GeoFrame's constructs and the respective GeoDB design methodology, a hypothetical support system for Brazilian agrarian reform will be used, described as follows.

The Brazilian government is initiating a process of land distribution for families of rural workers, in which nonproductive large landholdings are dispossessed to be divided and distributed. Each family of rural worker receives a parcel, size varying according to the country region and also depending on the available resources in the region, such as existent cropped areas, pastures, local roads, storage places, housing, or even natural resources availability such as water sources, streams, native vegetation, etc. A GIS application is used to assist in the demarcation of new parcels to be distributed. This distribution is carried out based on criteria that take into consideration, besides the

M



Modeling with a UML Profile, Figure 1 The GeoFrame framework. *TIN* Triangular irregular network resources previously mentioned, the relief, soil and vegetation type. Finally, effective environmental laws must be considered, as state laws prohibit the cultivation of agricultural crops in permanent protection areas (hill tops), areas with slope above 45° or close to water resources (lakes and rivers).

A short description of the classes belonging to each of the three GeoFrame abstraction levels follows. The way these classes are used during the modeling process is described in UML-GeoFrame Methodology section.

Planning Level This comprises:

- GeographicRegion: this defines the geographic regions corresponding to the interest areas of a GIS application. For instance, the region corresponding to a large property that will be divided and the region of the municipal district in which the property is located.
- Theme: each geographic region can be portrayed by several themes. Two examples of themes in the context of the agrarian reform system are Allotment and Environmental Aspects. The themes can be organized in a theme and subtheme hierarchy. Hence, the Environmental Aspects theme could contain, for example, the subthemes Relief, Vegetation and Hydrography.

Metamodel Level This comprises:

- ConventionalObject: this generalizes the classes of the application without spatial representation. An example could be the worker family class that will receive a parcel from the agrarian reform project.
- GeographicPhenomenon: this generalizes all application classes that define the geographic phenomena. This class is specialized in the GeographicField and GeographicObject classes.
- GeographicField: this class is a generalization of all application classes that are perceived in field view. These phenomena are also known as attributes varying in field, for example the classes Relief, Vegetation, Soil Use and Temperature.
- GeographicObject: this generalizes the application classes perceived in object view, in other words, classes whose instances have a single identity. Examples include Municipal districts, Farms, Parcels, Rivers and roads.

Spatial Representation Level The spatial representation level in the GeoFrame, as part of the conceptual data modeling process that prioritizes "what" rather than "how", allows designers and users to specify the type of spatial representation used to abstract the spatial characteristics of each geographic phenomenon.

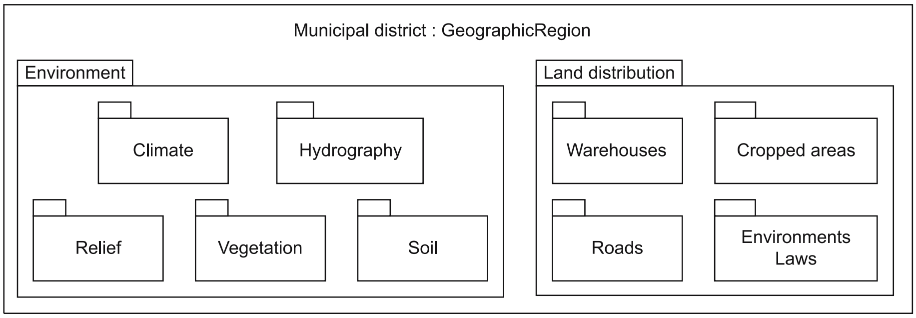
The purpose of GeoFrame is not to specify a type of data structure needed to store the spatial datum into the SDBMS, but only to model the spatial component of a particular geographic phenomenon as the user perceives or abstracts. For instance, in an urban water network application, the Hydrant class is associated with a spatial object of Point type (subclass of SpatialObject). This association only informs that the spatial characteristic of the geographic phenomenon hydrant will be zero-dimensional. However, as it corresponds to a vertex in a network structure, besides the X and Y coordinates, in SDBMS the vertex must be interlinked to other network elements (in order to maintain the topology) through a data structure of type arc-node. Nevertheless, this specification should only be detailed in the logical design phase of GeoDB. Following this approach, the GeoFrame classes in the spatial representation level are:

- SpatialObject: generalizes the classes of spatial representation of geographic phenomena in object view, such as Point, Line, Polygon or ComplexSpatialObject, which recurrently consists of two or more spatial objects. This last type of spatial representation is used when the geographic phenomenon presents a composed or complex characteristic (example: an archipelago).
- FieldRepresentation: the conceptual modeling of spatial representation of geographic phenomena in field view is the major difference of the UML-GeoFrame compared with others. The FieldRepresentation class generalizes the main types of spatial representation used to abstract the spatial characteristic of phenomena in field view, which are: GridOfCells, AdjPolygons, Isolines, GridOfPoints, triangular irregular network (TIN) and IrregularPoints. This specification only deals with the way the designer/user frequently abstracts the spatial form of geographic fields. For example, many users imagine the relief as a geographic phenomenon usually represented by isolines, although other users work with the relief represented by a TIN or a digital elevation model. This basic group of six spatial representation models for phenomena in field view was identified and described in [14] as containing the models most commonly found in GIS. However, new models of spatial representation for fields can be added to GeoFrame.

A project methodology for GeoDB is presented next. GeoFrame consists of a class library that provides the fundamental elements present in any GeoDB. Nevertheless, the basic classes of GeoFrame are represented in the conceptual data schema only in an implicit way, which is done through stereotypes, as shown below.

The UML-GeoFrame Methodology

A methodology for GeoDB modeling based on the UML-GeoFrame is described here; in other words, the steps that



Modeling with a UML Profile, Figure 2 Example of hierarchical theme diagram

should be followed during the GeoDB modeling process and how the GeoFrame elements are integrated with the UML class diagram constructs are presented.

The modeling process based on the UML-GeoFrame comprises five steps:

- Step 1: to identify themes and subthemes for each target region of the application
- Step 2: to draw a class diagram for each theme specified in step 1, associating classes of different themes, if this is the case
- Step 3: to model the spatial characteristic of each geographic phenomenon
- Step 4: to model spatial relationships
- Step 5: to model temporal aspects

The following subsections present each step in detail.

Step 1: to identify themes and subthemes for each geographic region GIS applications are usually developed focusing on a particular geographic region, it can be an area of great extension such as a country, a state, a municipal district or a large river basin. Maps at small scales are usually manipulated in these applications. On the other hand, numerous applications focus on smaller areas such as a city, neighborhood, farm or small river basins, in which the degree of granularity of the manipulated data is usually much higher, with data usually represented in large-scale maps.

The specification of various themes that will portray each geographic region, besides allowing a top-down approach to the problem, also aims at facilitating the understanding of large data schema.

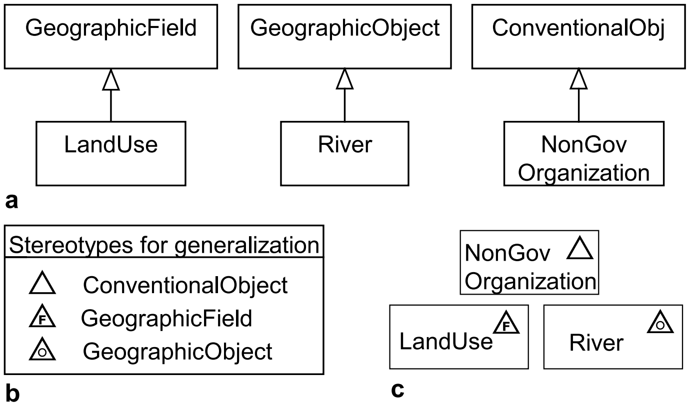
The elements identified at this stage are not directly transformed into a SDBMS structure; on the contrary, they are used only in the conceptual modeling phase, enabling the designer to plan and administer the data schema complexity. Theme modeling is done using the UML package construct. Figure 2 shows a possible hierarchical theme diagram for the agrarian reform support system.

Once the diverse themes are defined, the designer will be able to focus on a specific theme at a time to carry out data modeling (described in Step 2). For instance, one can chose Hydrograph and model all the classes of objects pertaining only to this theme. This process simplifies modeling and facilitates the understanding of the problem domain by the designer, as well as the communication with users.

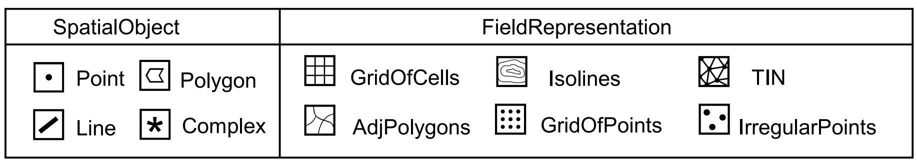
Step 2: to draw a class diagram for each theme At this stage, modeling of the data is carried out. For each theme, the several elements of the real world that is being modeled are abstracted. This stage is similar to the traditional database modeling, in which the essential elements of the reality are classified and modeled with the UML class diagram constructs. At this stage, based on the GeoFrame, three classes of objects are identified: conventional, geographic phenomena perceived in object view and phenomena in field view. The existent associations are also modeled among the classes, however, without considering the spatial constraints, which will be discussed in Step 4.

In a modeling process based on a class hierarchy, such as GeoFrame, the application classes should be modeled as subclasses of the GeoFrame classes (Fig. 3a). However, if all the application classes are represented as subclasses of only one of the three classes of the GeoFrame metamodel level, the data schema will be completely overloaded and difficult to read. Thus, three stereotypes were defined to replace these generalization–specialization relationships with graphic symbols (Fig. 3b), resulting in the representation shown in Fig. 3c. The idea of replacing relationships with graphic symbols was originally proposed by [15]. The advantage of using these three stereotypes is that in large data schemas, the designers and users can easily identify the application classes according to their main category.

Step 3: to model the spatial characteristic of each geographic phenomenon Geographic phenomena differ from conventional objects in that they have spatial properties (attributes and relationships), which are represented in a SDBMS by data structures containing primitive geometric object instances (e. g., point, line) or complex ones (e. g., multipoint, multipolygon), whose coordinates repre-



Modeling with a UML Profile, Figure 3 Generalization–specialization stereotypes



Modeling with a UML Profile, Figure 4 Stereotypes for spatial representation

sent points in the geographic space based on a cartographic projection system (e. g., UTM).

During the conceptual modeling stage, the designer should not be stuck on details such "how" the geographic phenomena will be stored in SDBMS, but only on abstracting its spatial characteristics. For example, a pole in an electric network has the spatial characteristic of a point, while a lake has the characteristic of extension in the form of a region/polygon. In turn, the relief, which is perceived as a geographic phenomenon in field view, can have its spatial characteristic abstracted by a spatial model of the isoline type. Moreover, the spatial characteristic of a geographic phenomenon can be abstracted from different cartographic representation models, characterizing the existence of multiple representations for the same geographic phenomenon.

At the beginning of the 1990s, based on the object-oriented paradigm, a number of modeling languages proposed that geographic phenomena were modeled as specialized subclasses of a set of predefined classes representing geometric objects. Therefore, the class Street would be modeled as a subclass of the class Line, a class Pole should be modeled as a subclass of the class Point, etc. There was in this approach an incorrect use of the generalization–specialization concept, in which two distinct things were related by an IS-A relationship. That is, a road is not a line and a line is not a road, although a geometric object Line can be related to a geographic object Road to represent its spatial location.

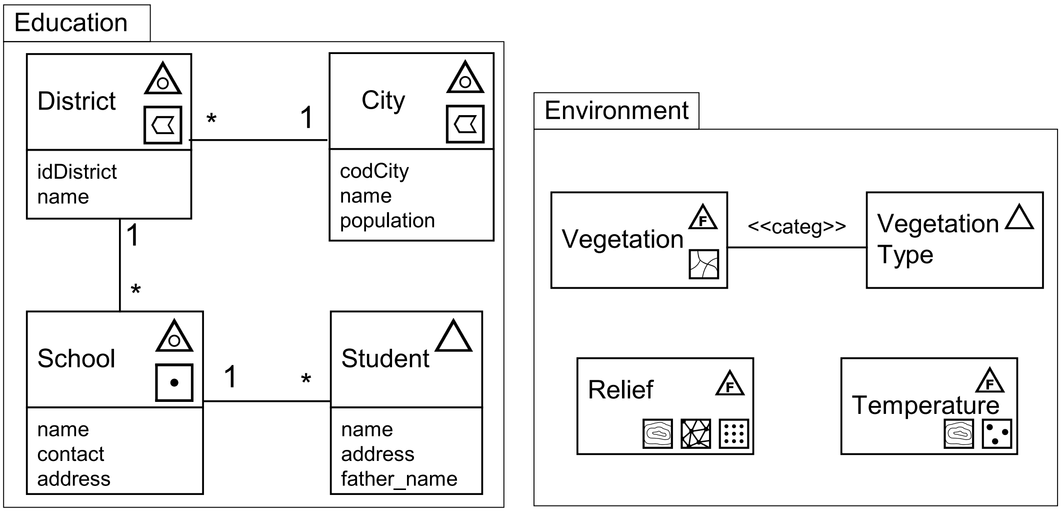
With the UML-GeoFrame, the spatial characteristic of geographic phenomena is not abstracted in the form of spatial attributes, but by means of associations between the classes of geographic phenomena and the classes of spatial representation. This is specified by the *represent* association in the GeoFrame (Fig. 1). Again, in order not to overload the data schema, stereotypes are defined to replace these associations (Fig. 4).

Therefore, in an UML-GeoFrame data schema, each geographic phenomenon class of the application domain will have at least two stereotypes, one for specialization and another for spatial representation. Figure 5 shows an example of a UML-GeoFrame data schema with two packages, one related to the Education theme and the other related to the Environment theme.

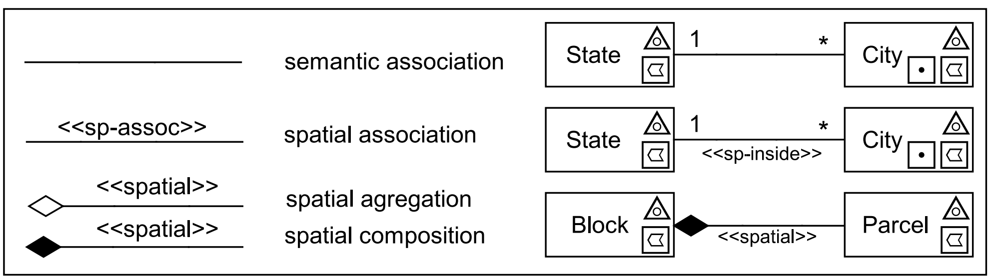
As can be seen in this example, the UML-GeoFrame enables a natural, integrated and consistent form of modeling phenomena perceived in field view. In this way, the conceptual data schema contains all the elements of the real world to be stored in the SDBMS and vice versa.

Step 4: to model spatial relationships A relationship modeled on a conceptual data schema semantically implies two facts; the first is that objects of the respective classes can be associated, which allows the user to consult the database based on this relationship, the second is that DBMS will have to guarantee the integrity constraint specified by the multiplicity of this relationship automatically. For example, an one-to-many association (1..* in UML) between the classes City and Farm, means that each farm must be associated at the most to one city, and that each city can be associated to zero or many farms.

In a GIS, this type of relationship can be captured in two ways: spatial and semantic. The spatial way is obtained by spatial operations between two geometric objects, for instance, verifying whether a polygon is inside another



Modeling with a UML Profile, Figure 5 Example of a data schema using the UML-GeoFrame notation



Modeling with a UML Profile, Figure 6 Stereotypes for spatial relationship

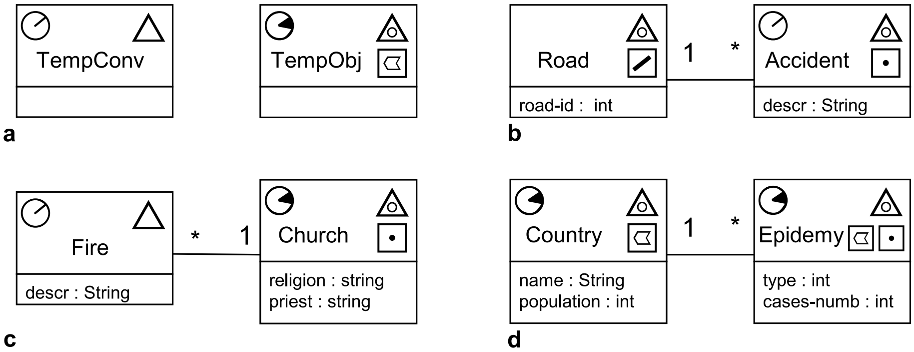
polygon. The semantic way is the traditional database way, where the farm related register contains a field that stores a reference (e. g., foreign-key) for the city that the farm is associated with.

With the UML-GeoFrame the designer can specify the two types of relationships (spatial and semantic), meaning that the SDBMS must guarantee the specified integrity constraint. The semantic relationships correspond to the associations normally specified between classes in the data schema. In this case, referential integrity constraints can be transformed, for example, to foreign-key constraints in a relational DBMS. The integrity constraints of spatial relationships must be implemented in GIS or SDBMS specific procedures.

The possible types of spatial relationships are topological or metric. Engenhofer et al. [16] described a set of types of spatial intersections that can occur between two regions: disjoint, contain, inside, equal, touch, covers, covered by and overlap. This set can be applied to other types of combinations such as point and polygon or line and polygon.

The specification of spatial associations in the UML-GeoFrame is done with textual stereotypes, i. e., a text between "...". The text corresponds to the type of spatial constraint one wants to impose, which can be of any type, including the relationships shown in Fig. 6. If there is no textual stereotype, then the association is semantic. Thus, in the logical design phase, the semantic association between State and City (Fig. 6) will be implemented as an attribute in the table City with foreign-key constraint for the table State, whereas the spatial association between State and City will be implemented by a procedure that will apply an *inside* spatial operation. These examples also show that an instance of City can have multiple spatial representations (point or polygon). Finally, UML constructs of aggregation and composition correspond to WHOLE-PART relationships. Figure 6 shows a spatial composition where a Block is modeled as the whole and the Parcels correspond to the parts.

Step 5: to model temporal aspects Numerous geographic phenomena are dynamic, i. e., their properties



Modeling with a UML Profile, Figure 7 **a–d** Examples of spatiotemporal classes

(descriptive and spatial) undergo changes with time. Although most databases applications reflect only the current state of real-world objects they represent, many applications need to keep the data evolution description as they are changed.

Descriptive, temporal and spatial dimensions are orthogonal. Temporal properties can therefore be defined for the three geospace data categories: conventional objects, geographic phenomena perceived in field view and geographic phenomena perceived in object view. In temporal DBMS, two types of time can be defined: valid time and transaction time. Rocha et al. [17] presents an extension to the UML-GeoFrame comprising these types of time. However, the enormous possibility of combinations between the different dimensions (e. g., temporal and spatial) led to a highly complex model that was really very little used by GIS users/designers. Thus, only the valid time is presented here.

Valid time is the time instant or time interval when an object of the real world is considered valid. For example, the strike against the World Trade Center took place on September 11, 2001; in turn, the Gulf War occurred in the period between August 1990 and February 1991. Hence, another important factor is the granularity of temporal information. UML-GeoFrame considers three types of time granularity: Date, Time and Timestamp. Specifying the granularity of a temporal attribute is the same as defining the domain of a descriptive attribute value (e. g., CHAR or Boolean).

Besides granularity, the designer can specify two types of temporal occurrence: *Interval* and *Instant*, which is done using the stereotypes shown in Fig. 7a. An *Interval* temporal class indicates the need for storage of its evolution, that is, if its properties (descriptive or spatial) are changed, a new version of the same object will be created. In this case, temporal attributes will be inserted to each object version, indicating the initial and final valid time of this object version. In the case of *Instant* temporality, the class will contain temporal attributes, but new versions will not be created for its objects. The UML-GeoFrame also allows the specification of temporal relationships. Following, some examples of modeling of temporal aspects are presented.

Figure 7b shows an association between the Road and Accident classes. The Road class has only spatial aspects, whereas the Accident class is spatiotemporal of the *Instant* type, characterizing the need for storing the instant (e. g., timestamp) when the accident took place. In the case of *Instant* temporality, each occurrence (a new accident) creates a new object, and therefore temporal versions of the object will not be generated.

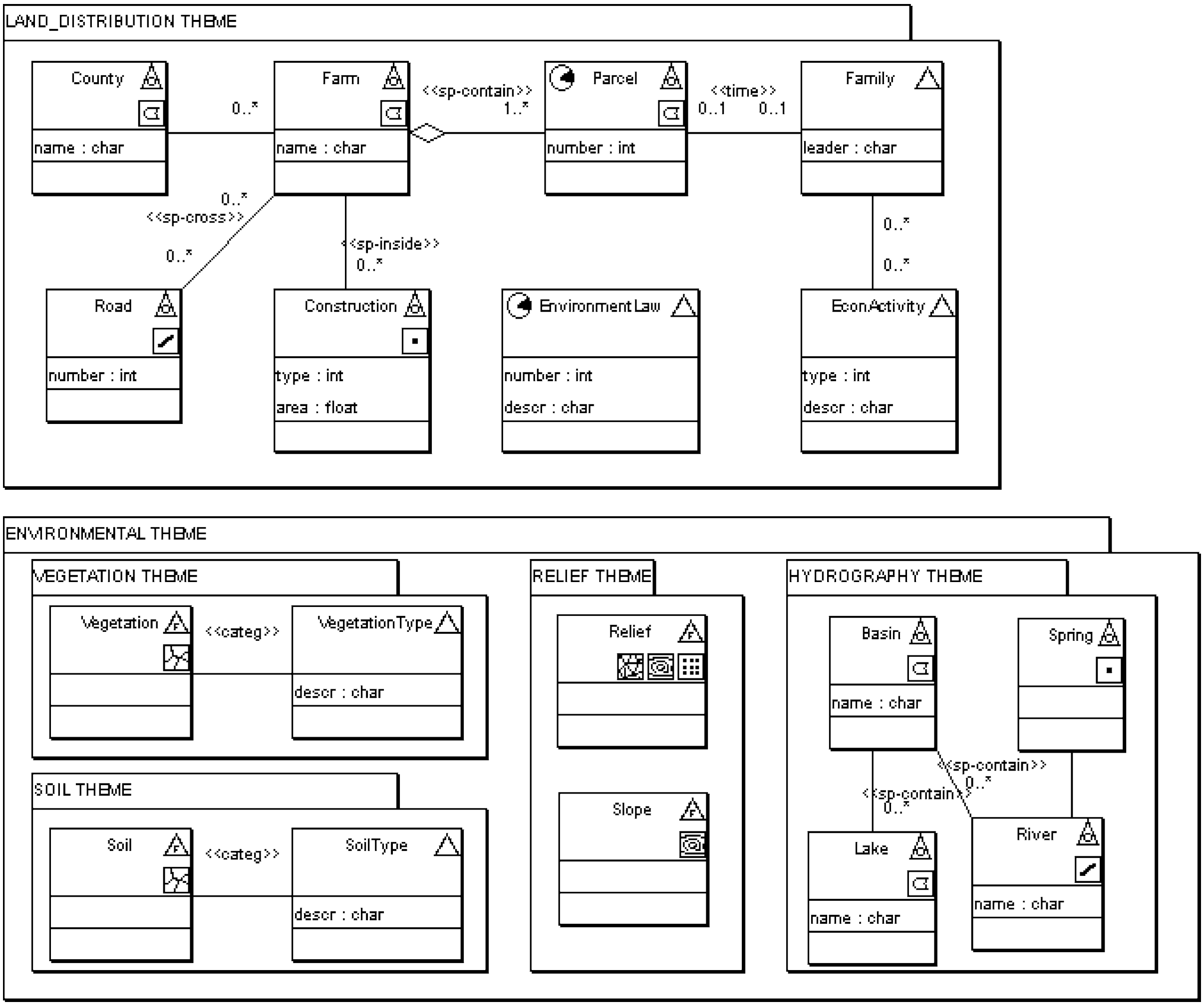
Figure 7c illustrates the Fire class, whose instances are conventional objects with *Instant* temporality, associated with a spatiotemporal class Church. Each fire occurrence generates a new object instance that contains temporal data recording fire valid time (e. g., date or year). In turn, each change in Church properties, with *Interval* temporality, creates a new version of the same object.

Figure 7d shows two spatiotemporal classes of the *Interval* type. Each time that a property of the Country class (e. g., population) is changed, a new version of the object country will be created. The same occurs with the class Epidemic, in which both the number of cases and the affected geographic region can be changed.

The UML-GeoFrame methodology establishes the five steps described, but they do not need to be executed necessarily in this order. A more experienced designer can model the three aspects (descriptive, spatial and temporal) of a class at the same time. Nevertheless, considering in detail all the aspects involving a GeoDB design is not a trivial task. The objective of the methodology is to assist the designer to work methodologically, in an incremental and organized way.

An Example of the Application for Brazilian Agrarian Reform

Figure 8 shows the final UML-GeoFrame diagram, relative to the hypothetical system of support to agrarian reform, described at the beginning of this entry. This



Modeling with a UML Profile, Figure 8 Final class diagram for the hypothetical system of support to agrarian reform

schema was depicted using the ArgoCASEGEO CASE tool [13].

In Fig. 8, how the division of the data schema using packages describing the diverse themes facilitates the understanding of the project is shown. This example also shows how natural is modeling, in an integrated way, conventional objects and geographic phenomena in field and object views. Multiple spatial representations, spatial relationships and temporal aspects are also easily specified using the UML-GeoFrame.

Key Applications

The development of a GIS application, however simple it might be, will bring great benefits if a conceptual data modeling language is used during the system design phase. The use of these conceptual data modeling languages in medium- and large-size systems is fundamental. GIS applications have become more and more integrated with other corporative information systems, sharing diverse data bases and running not only as operational control systems (e. g.: cadastres of properties and infrastructure service networks), but as important decision-making support systems.

Future Directions

A CASE tool named ArgoCASEGEO has been developed [13] to assist GeoDB modeling using UML-GeoFrame. This tool generates logical–spatial data schemas for the main data models related to GIS software (e. g., Shape File, Oracle Spatial). One of the innovative characteristics of this tool is to support a catalogue of analysis patterns [18] aimed at enabling the reuse of GeoDB design solutions by different designers.

Acknowledgements

This work has been partially supported by CNPq (Brazilian National Research Council) and Fapemig.

Cross References

► Modeling with Pictogrammic Languages

Recommended Reading

1. Shekhar, S., Chawla, S.: Spatial Databases: A Tour. Prentice Hall, New York (2003)
2. Elmasri, R., Navathe, S.B.: Fundamentals of Database Systems, 3rd edn. Addison-Wesley, Reading, MA (2000)
3. OMG – Object Management Group (2007) Unified Modeling Language. Available at http://www.uml.org
4. Bédard, Y., Larrivée, S., Proulx, M.J., Nadeau, M.: Modeling geospatial databases with plug-ins for visual languages: a pragmatic approach and the impacts of 16 years of research and experimentations on Perceptory. In: Wang, S. et al. (eds.) Conceptual Modeling for GIS (COMOGIS) Workshop ER2004, Shanghai, China, 8–12 Nov 2004. Lecture Notes in Computer Science, vol. 3289, pp. 17–30. Springer, Berlin (2004)
5. Borges, K.A., Davis, C.D., Laender, A.H.F.: OMT-G: An object-oriented data model for geographic applications. GeoInformatica **5**, 221–260 (2001)
6. Lisboa Filho, J., Iochpe, C.: Specifying analysis patterns for geographic databases on the basis of a conceptual framework. In: ACM Symposium on Advances in Geographic Information Systems, pp. 9–13. Kansas City, 5–6 Nov 1999
7. Couclelis, H.: People manipulate objects (but cultivate fields): beyond the raster-vector debate in GIS. In: Theories and Methods of Spatial-Temporal Reasoning in Geographic Space. Lecture Notes in Computer Science, vol. 639, pp. 65–77. Springer, Berlin (1992)
8. Chen, P.P.S.: The entity-relationship model: towards a unified view of data. ACM Trans. Database Syst. **1**, pp. 9–36 (1976)
9. Abiteboul, S., Hull, R.: IFO: A formal semantic database model. ACM Trans. Database Syst. **12**, 525–565 (1987)
10. Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W.: Object-Oriented Modeling and Design. Prentice-Hall, Englewood Cliffs (1991)
11. Coad, P., Yourdon, E.: Object-Oriented Analysis. 2nd edn. Prentice-Hall, New York (1991)
12. Booch, G., Jacobson, I., Rumbaugh, J.: The Unified Modeling Language User Guide. Addison-Wesley, Reading, MA (1998)
13. Lisboa Filho, J., Sodré, V.F., Daltio, J., Rodrigues, M.F., Vilela, V.: A CASE tool for geographic database design supporting analysis patterns. In: Wang S. et al. (eds.) Conceptual Modeling for GIS (COMOGIS) Workshop ER2004, Shanghai, China, 8–12 Nov 2004. Lecture Notes in Computer Science, vol. 3289, pp. 43–54. Springer, Berlin (2004)
14. Goodchild, M.F.: Geographical data modeling. Comput. Geosci. **18**, 401–408 (2002)
15. Bédard, Y., Paquette, F.: Extending entity/relationship formalism for spatial information systems. In: AUTO-CARTO 9, Ninth International Symposium on Computer-Assisted Cartography, ASPRS-ACSM, Baltimore, USA, 2–7 April 1989
16. Engenhofer, M., Clementini, E., Felice P.: Topological relations between regions with holes. Int. J. Geogr. Inf. Syst. **8**, 129–144 (1994)
17. Rocha, L.V., Edelweiss, N., Iochpe, C.: GeoFrame-T: a temporal conceptual framework for data modeling. In: ACM Symposium on Advances in Geographic Information Systems, Atlanta, 9–10 Nov 2001, pp. 124–129
18. Lisboa Filho, J., Iochpe, C., Borges, K.A.: Analysis patterns for GIS data schema reuse on urban management applications. CLEI Electron. J. **5**, 1–15 (2002)

Modeling with Enriched Model Driven Architecture

ANDRÉ MIRALLES[1], THÉRÈSE LIBOUREL[2]

[1] Centre for Agricultural and Environmental Engineering Research, Earth Observation and GeoInformation for Environment and Land Development Unit, Montpellier Cedex 5, France

[2] University of Montpellier II – National Center for Scientific Research, Montpellier Laboratory of Computer Science, Robotics, and Microelectronics, Montpellier Cedex 5, France

Synonyms

MDA; Model driven architecture; Model driven development (MDD); MDE; Model driven engineering

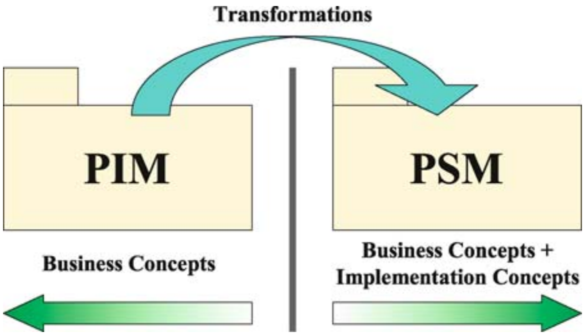
Definition

Model Driven Architecture, formalized in 2001, is a software design approach proposed by the Object Management Group (OMG) with the objective of improving application development. It was conceived of in order to improve the productivity of software development but also to resolve problems of software portability, software integration and software interoperability encountered during development [4].

To achieve this objective, the MDA approach recommends separating the specification of system functionality from the specification of the implementation of that functionality on a specific technology platform. For that, the authors of this approach suggest the use of two types of model groups: the Platform Independent Models (PIM) and the Platform Specific Models (PSM).

PIMs are models providing a description of the structure and functions of a system without technical specifications of data-processing nature. PSMs are models defining how structure and functions of a system are implemented on a specific platform.

In fact, the MDA approach introduces a separation between concepts and specifications needed to develop software. PIMs only contain business concepts. PSMs contain business concepts and also implementation concepts. Since all of the PIM business concepts are includ-



Modeling with Enriched Model Driven Architecture, Figure 1
Concepts separations and transformations

ed in PSMs, a PIM can be seen as a modified subset of a PSM. Therefore, a PSM always derives from a model PIM through one or more transformations. Fig. 1 illustrates this separation and transformation. If different platforms are used for the implementations (e. g., same standardized model implemented into different organizations), then more than one PSM may be derived from the same PIM.

The previous transformations, called PIM/PSM transformations, are not the only ones. In fact, the authors of MDA mention, on the one hand, the existence of PSM/PIM transformations converting a PSM into PIM and, on the other hand, transformations whose models source and target are in the same fashion standard: PIM/PIM transformations or PSM/PSM transformations.

In the process of development, the PSM is not the last step since it is then necessary to project this model into a programming language. This projection is often considered to be a transformation.

In summary, the separation introduced by MDA can be seen as a solution of a more fundamental preoccupation: the capitalization of knowledge.

Historical Background

The Model Driven Architecture is an approach which expands the Object Management Architecture [5] without replacing it. The Object Management Architecture provides a framework to implement distributed systems [6] whereas the Model Driven Architecture defines an approach specifying how to use the models during the development of an application.

The founding text of Model Driven Architecture is a recent OMG work that was approved in 2001 [5]. In 2003, this text was supplemented by a methodological guideline defining the main directives to apply this approach.

In 2003, Anneke Kleppe et al. wrote in their book entitled, *MDA Explained: The Model Driven Architecture-Practice and Promise* [4], that this approach was "still in its infancy". The following year, Anneke Kleppe, when interviewed about Model Driven Architecture [3], was quoted as saying, "Can we realistically look to build all of the J2EE code for an application from UML diagrams?" and "Yes, it may take five, ten, or maybe twenty years before it will be common practice, but I believe it will become just that."

These two statements explain that a lot of research has to be carried out so that this approach reaches its maturity. For over 20 years, R&D and commercial software engineering software packages have offered semi-automated transformations between so-called "conceptual" and "physical" models, with more or less success with regards to their widespread usage. Nowadays, with the standardization impacts of MDA and UML, hope is returning. Currently, research works are mainly concentrated on the transformations of models to automate these transformations in order:

- Firstly, to carry out a *Full* MDA process [3], i. e., a process automating the evolution of the models from the analysis to the implementation
- Secondly, to increase the productivity during the development and facilitate code reuse.

Scientific Fundamentals

The Enriched Model Driven Architecture is a framework for designing and implementing spatio-temporal databases following a MDA approach into a Full MDA process. This process includes the generation of the SQL code necessary to implement the spatio-temporal database. For that, a multimodel artifact and a panoply of transformations has been conceived and implemented.

In order to describe the spatial and temporal properties of the business concepts, the pictogrammic language of the Perceptory case tool [1] has also been adopted and implemented.

Principle of the Software Development Process Model (SDPM)

When an application is developed, one of the main preoccupations for a project manager and for the company in charge of the software development is the capitalization and reuse of the knowledge accumulated during the development.

The capitalization of knowledge is not just the problem of separating the business concepts and implementation concepts according to the MDA vision. *The capitalization of knowledge is problematic throughout the development of an application.* The *Software Development Process Approach* (SDPA) is a framework based on this report [7].

As with each phase of the development, the type of mobilized knowledge is different. The idea of this new approach is to capitalize the knowledge of each one of the phases. For that, this new approach recommends dedicating a model at each one of the phases.

In order to materialize this new approach, a multimodel artifact, called the *Software Development Process Model* (SDPM), has been conceived [7]. This multimodel artifact contains the different models corresponding to the phases of the software development process. In this vision, the Software Development Process Model is the MODEL of the application under development. Fig. 2 shows the Software Development Process Model for the development of software following the Two Track Unified Process method [8] derived from the Unified Process method. This figure also shows that the PIM/PSM separation introduced by MDA occurs when the project moves from the advanced design phase to the implementation phase.

Panopoly of Transformations for a Full MDA Process [7]

Diffusion Transformation and Management of the Software Development Process Model In order to manage the different models, the Software Development Process Model has been endowed with a Diffusion transformation. This transformation first clones a concept from a source model into the following model. Step by step, the concepts that are captured in the analysis phase and added into the analysis model are transferred into the implementation models.

In order to guarantee the coherency of the multimodel artifact, a Cloning Traceability Architecture is automatically built by the Diffusion transformation. After cloning, this transformation establishes an individual cloning traceability link between each one of the source concepts and the cloned concepts. Fig. 3 illustrates the Cloning Traceability Architecture.

In an iterative development process, the Diffusion transformation adds, at every iteration, a new clone of the same source into the following model. To avoid this problem, when an individual cloning traceability link exists, the Diffusion transformation does not clone the concepts, but carries out only one update of the clone.

Full MDA Process for GIS Design and Development After a thorough analysis, the pictogrammic language of Perceptory [1] has been adopted to describe the spatial properties (Point, Line, and Polygon) and temporal properties (Instant and Period) of the business concepts. This pictogrammic language is applied on the analysis model of the SDPM. These pictograms are added to the business concept (i. e. classes) via stereotypes, one of extensibility mechanisms in UML. This mechanism allows to associate an icon or a pictogram on business concepts.

In the analysis model, the stereotype/pictogram couple only has an informative value. To be able to generate an application's code, these pictograms have to be reified into UML modeling elements. To do this, two geomatic transformations have been designed and implemented: the first generates a design pattern[1] based on spatial concepts and temporal concepts (Instant and Period) and the second converts the stereotype/pictogram couple into a relationship.

GIS Design Pattern Generation Transformation The spatial and temporal concepts have stable relationships that are entirely known. They constitute recurrent mini-models having the design pattern properties. Fig. 4 shows an example of a design pattern of the Geographic Information domain. The set of these patterns are called GIS Design Patterns.

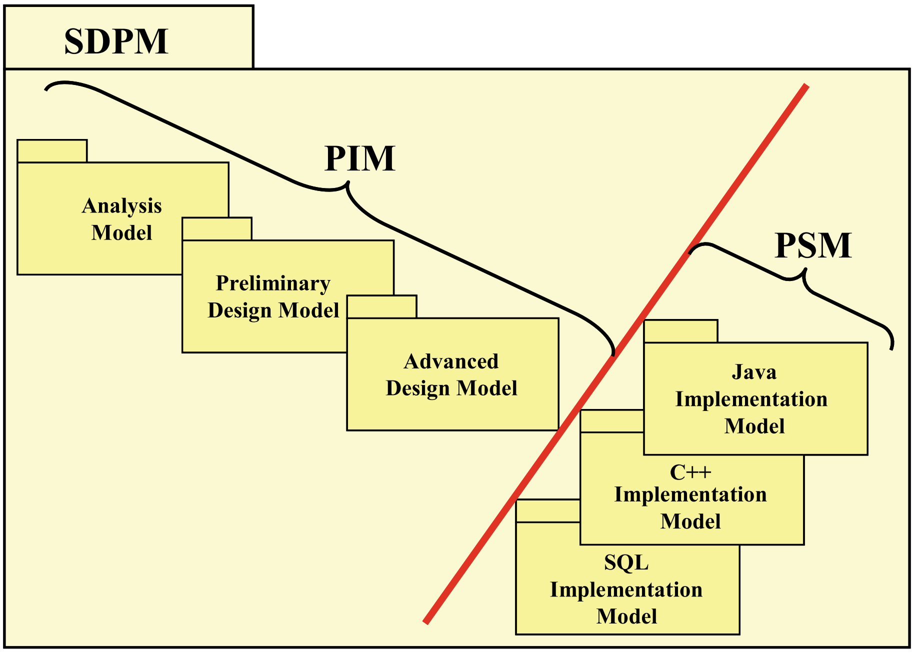
Given that the design patterns are always identical, they can be automatically generated without any difficulty. The GIS Design Pattern Generation transformation is the transformation in charge of generating the GIS Design Pattern.

Pictogram Translation Transformation Once the GIS design patterns have been created, the business and the spatial or temporal concepts represented by the pictogram are totally disassociated (Fig. 5). The goal of the Pictogram Translation Transformation is to automatically establish a relationship between the Parcel and Polygon concepts (cf. Fig. 5). This transformation creates an association called Spatial Characteristic.

During the capture of the pictogram, two tagged values are added to the business concept in order to specify the role of the spatial concept ({Gis S: Spatial Role(Geometry)}) and its cardinality ({Gis S: Spatial Cardinality(1)}). By default, this role and this cardinality have the values Geometry and 1, respectively, but the designer can subsequently modify them. In this association, the entity name has been allocated to its role, Parcel in this example, and its cardinality's value is 0..1. Once the association has been created, the stereotype/pictogram and the two tagged values are deleted since this information becomes redundant.

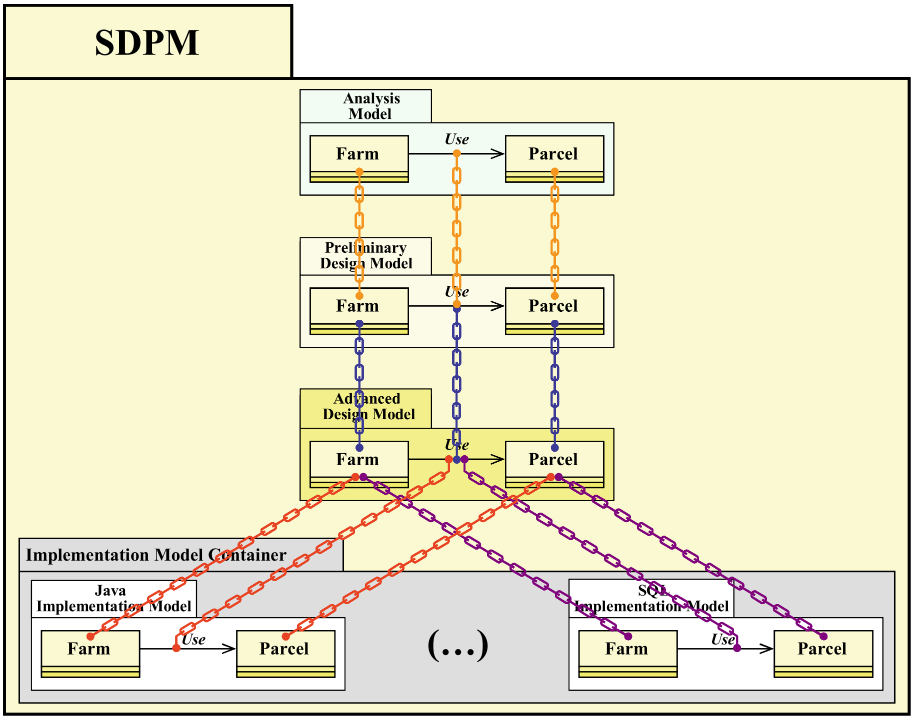
To ensure traceability, the Pictogram Translation Transformation creates a traceability link, called Translation Trace-

[1] A design pattern systematically names, motivates, and explains a general design that addresses a recurring design problem in object-oriented systems. It describes the problem, the solution, when to apply the solution, and its consequences. It also gives implementation hints and examples. The solution is a general arrangement of objects and classes that solve the problem. The solution is customized and implemented to solve the problem in a particular context [2]

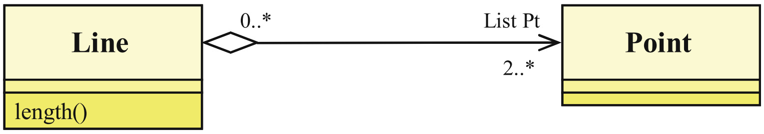


Modeling with Enriched Model Driven Architecture, Figure 2 Principle of the multimodel artifact Software Development Process Model

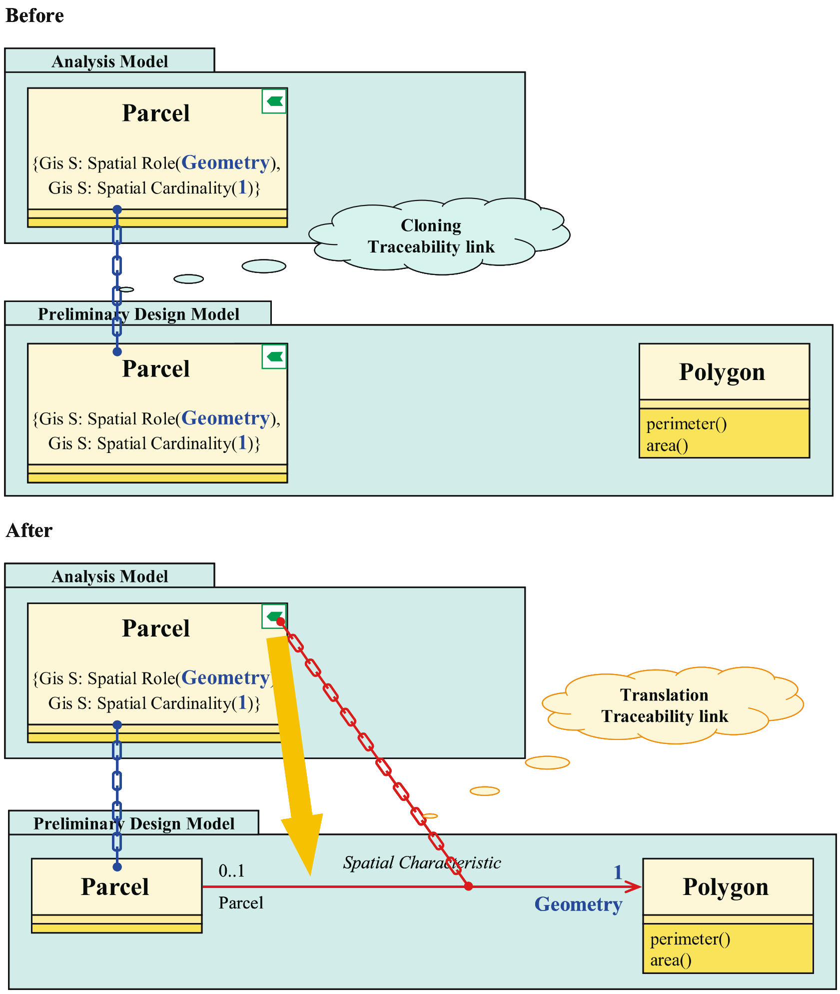
M



Modeling with Enriched Model Driven Architecture, Figure 3 Example of Cloning Traceability Architecture (to keep the figure simple, differing details of the models have been discarded)



Modeling with Enriched Model Driven Architecture, Figure 4 Example of a GIS Design Pattern



Modeling with Enriched Model Driven Architecture, Figure 5 Rule for transforming the stereotype/pictogram couple into an association

ability Link, between the pictogram of the business entity of the analysis model and the Spatial Characteristic association.

SQL Transformations In order to achieve a Full MDA process, SQL Transformations have conceived and implemented. They are applied on the SQL Implementation model. The objective of these transformations is to adapt the SQL Implementation Model after cloning to the SQL code generator of the Case Tool.

The main SQL transformation adds the persistence and primary key attributes (main features of relational database) on UML concepts. Annotated with information specific to SQL, it is possible to generate SQL code to create the spatial database. For that, the SQL code generator available in Case Tool supports is used.

Conclusion

The Software Development Process Approach is a recent contribution to the design and development of spatial databases. It enriches the MDA by subdividing the PIM level into three levels based on the development process. The reification of this approach is the Software Development Process Model, a multimodel artifact that meets the previously stated requirements of capitalization. This is so because each of the four modeling levels include concepts that are their very own and independent of those of the preceding level. Without the cloning traceability architecture and the Diffusion Transformation, the functioning and the maintenance of coherence of the Software Development Process Model would become impossible.

Structuring models into SDPM offers a very powerful mutation capability; it becomes very easy to change the hardware or software platform in the course of the development and to improve performance calculations, for example. Such changes can be made at a minimum cost since the analysis, preliminary design and advanced design models constitute the application's **capital**, capital that is available and mobilizable at any time. Thus, only the new implementation concepts will require reworking within the new implementation model. This mutation capability can also be called upon during the application's life cycle to correct bugs, improve features, mitigate obsolescence of hardware or software platforms, to create new versions of the application, etc.

The two geomatic transformations that have been designed use spatial, temporal and spatio-temporal properties of business concepts annotated with Perceptory pictogrammic language. Some SQL transformations adapt the SQL Implementation Model to the code generator of the Case Tool.

This panoply of transformations automates, for the first time in a full MDA process, the generation of SQL code that allows the Geographical Information System's database to be reified.

Key Applications

Software Development and Information Technology

The potential benefits in terms of productivity, portability, integration and interoperability confer a transversality that should be of interest to the software industry regarding the Software Development Process Approach; more so because this approach can be implemented irrespective of the typology of the software application to be created.

Geographic Information Systems and Spatial Databases

The field of Geographic Information Systems, including the design of associated spatial or spatio-temporal databases, benefit from Enriched Model Driven Architecture.

Future Directions

MDA is an approach that is still young; its broad principles were only enunciated by OMG in 2001. It will surely evolve in the coming years, either at the generalization level – the present solution is an example – or at the level of transformation languages (QVT, ATL, etc.).

Acknowledgements

The authors thank Professor Yvan Bédard of the Geomatics Research Center at Laval University and his team for stimulating our thought processes and for contributing to our research.

Cross References

► Computer Environments for GIS and CAD
► Patterns in Spatio-temporal Data

Recommended Reading

1. Bédard, Y., Larrivée, S., Proulx, M.J., Nadeau, M.: Modeling Geospatial Databases with Plug-ins for Visual Languages: A Pragmatic Approach and the Impacts of 16 Years of Research and Experimentations on Perceptory. In: Proceedings of ER Workshops 2004 CoMoGIS, Shanghai (China), November 2004
2. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design patterns – Elements of Reusable Object-Oriented Software, 1st edn., Addison Wesley Professional, London (2001)
3. Kleppe, A. (2004): Interview with Anneke Kleppe. Code Generation Network. http://www.codegeneration.net/tiki-read_article.php?articleId=21. date cited: August 2006
4. Kleppe, A., Warmer, J., Bast, W.: MDA Explained: The Model Driven Architecture-Practice and Promise. Addison-Wesley Professional, London (2003)
5. Miller, J., Mukerji, J. (2001): Model Driven Architecture (MDA). OMG. http://www.omg.org/cgi-bin/apps/doc?ormsc/01-07-01.pdf. date cited: September 2004
6. Miller, J., Mukerji, J. (2003): MDA Guide Version 1.0.1. OMG. http://www.omg.org/cgi-bin/doc?omg/03-06-01
7. Miralles, A.: Ingénierie des modèles pour les applications environnementales, Université Montpellier II, Montpellier, Dissertation (2006)
8. Roques, P., Vallée, F.: UML en Action – De l'analyse des besoins à la conception en Java, 2nd edn. Eyrolles, Paris (2002)

Modeling with ISO 191xx Standards

JEAN BRODEUR[1], THIERRY BADARD[2]
[1] Natural Resources Canada, Centre for Topographic Information, Sherbrook, QC, Canada
[2] Centre for Research in Geomatics (CRG), Université Laval, Quebec, Canada

Synonyms

Conceptual modeling of geospatial databases; Modeling geospatial databases; Modeling geospatial application database

Definition

Application Schema

ISO19109—Rules for application schema [1] defines an *application schema* as a conceptual schema for data required by one or more applications. In the context of geographic information, an application schema documents the content and the structure of geographic databases along with manipulating and processing operations of the application to a level of details that allows developers to set up consistent, maintainable, and unambiguous geographic databases and related applications [2]. As such, an application schema contributes to both the semantics of geographic data and describes the structure of the geographic information in a computer-readable form. It also supports the use of the geographic data appropriately (i. e., fitness for use). Typically, an application schema is depicted in a formal conceptual schema language.

Feature Catalog

As in ISO19110—Methodology for feature cataloguing [3], a *feature catalog* presents the abstraction of reality represented in one or more sets of geographic data as a defined classification of phenomena. In the context of geographic information, it is a collection of metadata that provides the semantics and the structure of the objects stored in a geographic database. A feature catalog includes (1) the names and definitions of feature types, (2) their properties' name and definition including feature attributes, geometry (shapes and specifications, datum, map projection, etc.), temporality (dimensions and specifications, datum, units, resolutions, etc.), operations, and roles, (3) descriptions of attribute values and domains, relationships, constraints, and so on. An application schema may be described in various forms such a text document, a database, a spreadsheet, etc. Typically, a feature catalog is available in electronic form to support interoperability of geographic information. Although both an application schema and a feature catalog address the same content basically, they are complementary in the manner they represent it.

Historical Background

International standardization in the geographic information field has taken place during the last decade. These works have been coordinated by the ISO Technical Committee 211 (ISO/TC 211) [4] and the Open Geospatial Consortium. (OGC) [5]. Standardization in geographic information aims at (1) facilitating and increasing the understanding and use of geographic data, (2) increasing the availability, the access, the integration, the exchange, and the sharing of geographic data (i. e., interoperability of geographic information), (3) enhancing efficiency when using geographic data, and (4) developing a worldwide orientation of geographic information in support of global problems (ecological, humanitarian, sustainable development, etc.) [4].

Scientific Fundamentals

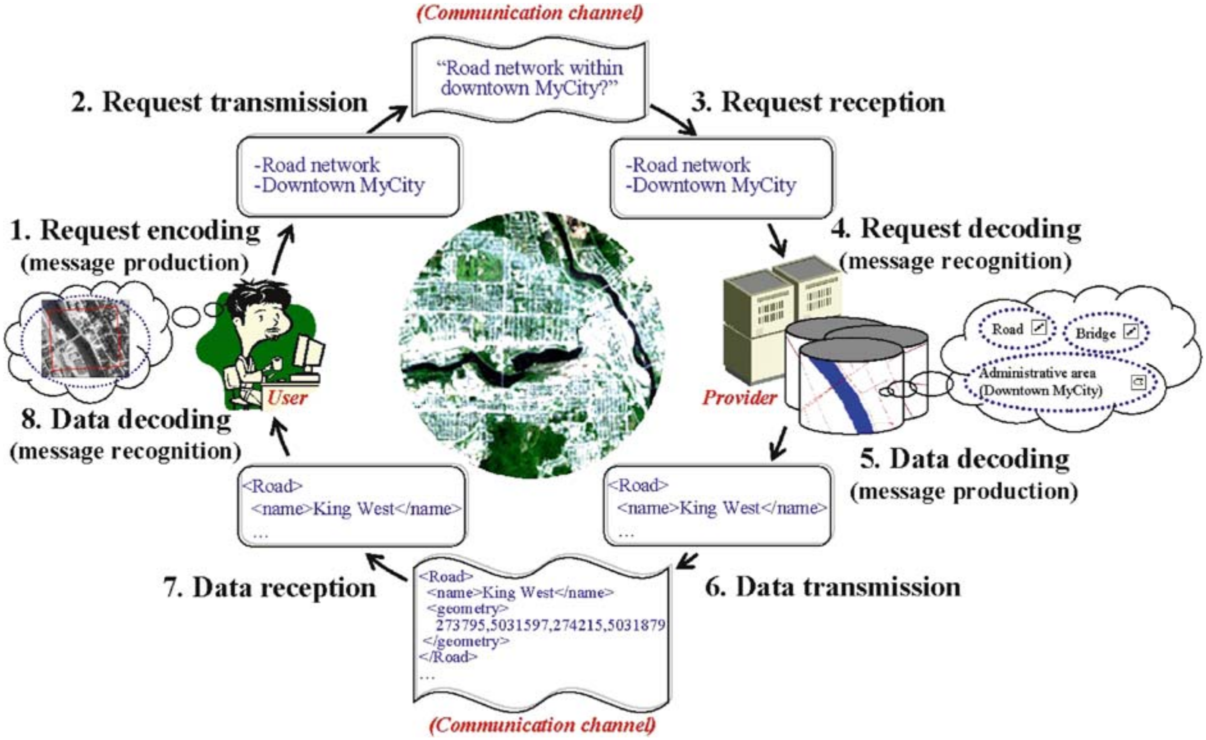
Interoperability

As introduced in [6], interoperability can be seen as the ability of two or more systems or components to exchange information and to use the information that has been exchanged. As such, interoperability adheres to the human communication process (see Fig. 1) where agents (e. g., human beings, systems, etc.) interact together at the system, syntactic, schematic, and semantic levels to share information. Each agent has its own conceptual representation of reality and uses it to encode (Fig. 1, steps 1 and 5) and decode messages (e. g., queries and responses about geographic information, Fig. 1, steps 4 and 8), which are transmitted (Fig. 1, steps 2 and 6) to or received (Fig. 1, steps 3 and 7) from another agent through the communication channel. Interoperability happens only when both agents engaged in a communication have the same understanding about the message [7]. Therefore, interoperability agrees to a bidirectional communication process including a feedback mechanism in both directions to control the good reception and understanding of messages.

Accordingly, topics of interest to standardization in geographic information are organization, content, access and technology, and education. Standards cover methods, tools, and data management services (including data definition and description) that are related to interoperability in geographic information including data acquisition, processing, analysis, access, portrayal, and transfer of data between users, systems, and places. They provide a structure for application development related to geographic data and refer to existing standards in information technology and communication (W3C (World Wide Web Consortium), OMG (Object Management Group), IETF (The Internet Engineering Task Force), OASIS (Organization for the Advancement of Structured Information Standards), and so on) when needed. A large number of standards are published as part of the ISO191xx suite of standards (see http://www.iso.ch/iso/en/stdsdevelopment/tc/tclist/TechnicalCommitteeDetailPage.TechnicalCommitteeDetail?COMMID=4637).

As ISO/TC 211 and OGC share common objectives, a similar program and have complementary approaches, they are now collaborating closely. On the one hand, the OGC wants their specifications to obtain official recognition of international standards. On the other hand, ISO/TC 211 wants to benefit from OGC's work and that OGC's specifications comply with ISO 191xx standards. By sharing certain of their resources and developing standardization projects jointly, they aim at reducing the inconsistency between *de jure* (i. e., official, approved by an official standardization body like ISO) and *de facto* or industrial standards (i. e., adopted by users, industry and/or professional sector, because of its popularity and its market share). As an example, ISO 19115—Metadata has been adopted by OGC as its Topic 11—Metadata, and the specifications of the Web Map Service (WMS) edited by OGC are now an ISO international standard (ISO 19128).

As such, the first part of this chapter concerns the role of conceptual modeling in the field of geographic infor-



Modeling with ISO 191xx Standards, Figure 1 Framework for interoperability

mation related to interoperability, while the second concerns the contribution of standards with respect to conceptual modeling in geographic information, before concluding remarks and future considerations are presented.

Geographic Information Modeling and Interoperability

Typically, organizations have assembled geographic databases for their explicit needs (e. g., censuses, land inventory and management, homeland security, sustainable development, etc.). Over time, a huge amount of geographic information has been accumulated. Considering the high demand for geographic information and the enhancement of web technologies, an increasing number of organizations have begun to disseminate of data they have collected. It is now well known that geographic information is widely available from multiple providers (i. e., governmental agencies, private organizations, etc.) and accessible on the web. Spatial data infrastructures (SDIs) have grown rapidly and, today, support easy access and use of geographic information (e. g., NSDI in the US, CGDI in Canada). Currently, those searching for geographic information can rely on SDIs, which offer a one-stop shop for geographic information in many countries.

However because the variety of available geographic information disseminated and further used without any formal definition, users experience problems in finding and obtaining the data that fit their needs, in analyzing such data, and in making sound decisions in different situations, thus limiting interoperability. Consequently, knowledge about data has become vital.

Knowledge about geographic information is collected in term of metadata. Metadata has been traditionally defined as "data about data" [8] and constitute a description of captured or modeled data in databases or applications. Basically, metadata refers to the content, the structure, the semantics, the lineage (source, collecting process, etc.), the quality (positional and content accuracy, etc.), the vintage, the resolution, the distribution format, the persons or institutions responsible for the data, etc. Accordingly, conceptual models, conceptual schemas, application schemas, database models, data dictionaries, feature catalogs, and repositories consist of so many approaches to document metadata specifically about the content and definition, the description of the structure, and the semantics of the geographic information. Considering the availability and accessibility to such descriptions of geographic information in a standardized manner, users of geographic information can interpret the information correctly and then identify the suitability of the data for their specific application (i. e., fitness for use). Therefore, metadata and especially conceptual models are crucial to support interoperability of geographic information.

Moreover in term of database design, it is an excellent practice to define properly the semantics, the geometry, the temporality, and integrity constraints of objects to be included in geographic databases and datasets. Referring to the interoperability framework depicted in the previous section, the content description of geographic databases consists in a fundamental agent's knowledge for reasoning and communicating with other agents, since it serves as its ontology. Ontology consists in a formal representation of phenomena with an underlying vocabulary including definitions and axioms that make the intended meaning explicit and describe phenomena and their interrelationships [7]. Database modeling benefits greatly from standards. Amongst existing standards, the ISO191*xx* suite of standards addresses specifically geographic information. The following section describes these standards, which contribute to the development of geographic database models and applications and to the standardized description of geographic information.

Contribution of Standardizations in Geographic Information to Spatiotemporal Modeling

Standards in geographic information contribute at various levels to the modeling of geographic information. The contributions range from the definition of types for data, which support the description of features and attributes, to the definition of standardized methodologies for handling descriptions of concepts of geographic databases or applications.

More specifically, the following standards take part either directly or indirectly to the modeling of geographic information:

- ISO/TS 19103 Conceptual Schema Language
- ISO 19107 Spatial Schema
- ISO 19108 Temporal Schema
- ISO 19109 Rules for Application Schema
- ISO 19110 Methodology for Feature Cataloging
- ISO 19111 Spatial Referencing by Coordinates
- ISO 19112 Spatial Referencing by Geographic Identifiers
- ISO 19115 Metadata
- ISO 19135 Procedures for Item Registration

This section is a review of the above standards' contributions to the modeling of spatiotemporal information. Hereafter, the expression "application schema" is used to refer to conceptual model, database model, or any other similar expressions. For a presentation of database modeling levels of abstraction, see the chapter entitled "Spatial Database Modeling Pictogrammic Languages."

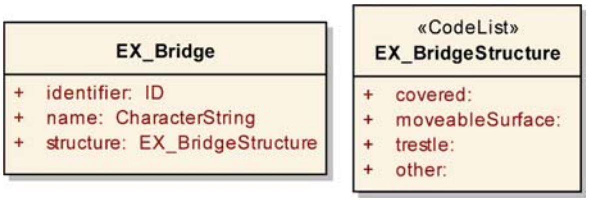
Conceptual Modeling Language Although ISO/TS 19103 Conceptual Schema Language [9] is intended to provide rules and guidelines for the use of a conceptual modeling language in ISO 191xx geographic information standards, namely the Unified Modeling Language (UML), it also includes definitions of a number of data types which provide a common ground for the representation of attributes and values.

These rules are a help to the harmonization of conceptual models. Basically, ISO/TS 19103 rules follow standard UML [10], but introduce restrictions on the definition of classes, use of multiple inheritance, association role and multiplicity, and multiple class association. For instance, ISO/TS 19103 specifies a naming convention to increase readability and consistency: each word of class and relationship names are capitalized without space between them; each word of attribute, relation role, operation, and parameter names are capitalized except the first, without spaces between words. All examples in this chapter follow ISO/TS 19103 rules.

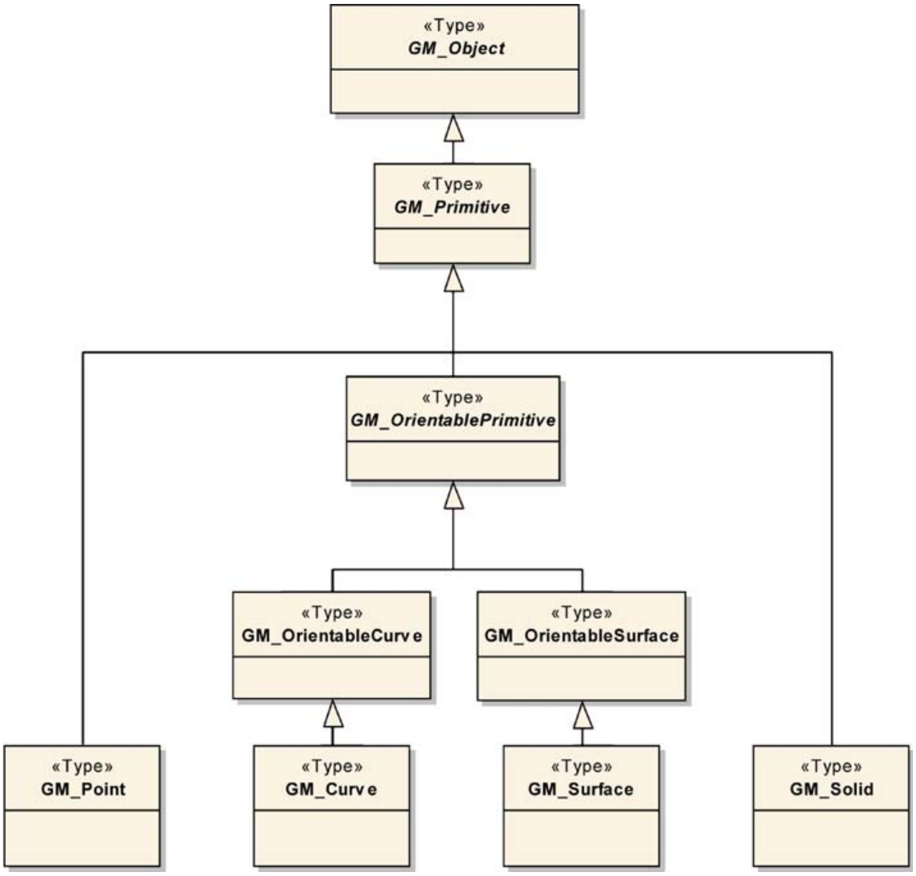
New stereotypes are also introduced: "CodeList", "Leaf", and "Union". "CodeList" is basically an extensible enumeration of character string values that an attribute can take. As Fig. 2 illustrates, the class *EX_Bridge* has an attribute *structure* of the type *EX_BridgeStructure*, which is a CodeList that enumerates its acceptable values, but is not limited to them. "Leaf" identifies packages that have no subpackages. A "Union" class is a class that gives to users a choice between multiple alternatives for the description of an attribute, but the attribute must use one, and only one, choice.

In addition, the use of standardized data types when building application schemas for geographic information supports attributes' meaning and better understanding of geographic information and, hence, enhances interoperability of geographic information. In ISO/TS 19103, these data types include primitive types, implementation and collection types, and derived types. Primitive types are those that are basic for the representation of elementary values. More specifically, they cover types for the representation of numeric, text, date and time, truth, multiplicities, and elementary enumerations:

- Numeric: Number, Decimal, Real, Integer, Unlimited-Integer, and Vector



Modeling with ISO 191xx Standards, Figure 2 CodeList example



Modeling with ISO 191xx Standards, Figure 3 Basic geometric primitives specified by ISO 19107 [11]

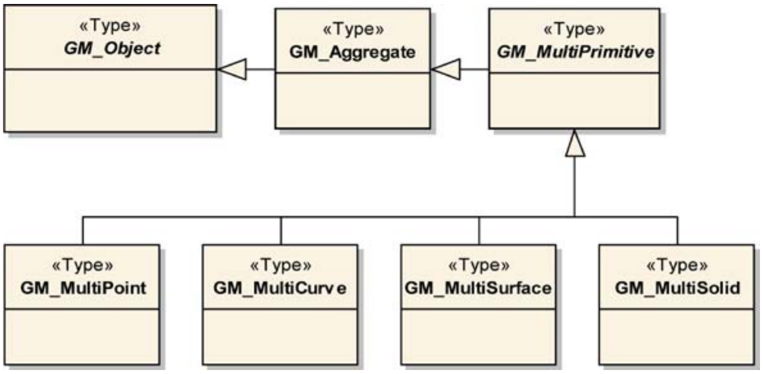
- Text: Character, Sequence<Character>, and CharacterString
- Date and time: Date, Time, and DateTime
- Truth: Boolean {TRUE =1, FALSE = 0}, Logical {TRUE = 1, FALSE = 0, MAYBE = 0,5}, and Probability
- Multiplicity: Multiplicity, MultiplicityRange
- Elementary enumerations: Sign {+, −}, Bit {0, 1}, Digit {zero = 0, one = 1, two 2, three = 3, four = 4, five = 5, six = 6, seven = 7, eight = 8, nine = 9}

A collection type is a container of multiple instances of a given type. Collections may have different characteristics with respect to ordering, duplication, and operations. Four types of collection are defined: *Set*, *Bag*, *Sequence*, and *Dictionary*. A *Set* consists of a definite number of objects of a given type. In a *Set*, an object appears once and only once (no duplicate are allowed) and objects are not ordered. A *Bag* is a similar structure to *Set*, but accepts duplicates. A *Sequence*, commonly known as a List, is a structure similar to a *Bag* in which objects are ordered. The *CircularSequence* is a specific type of *Sequence*, which does not define any last element making it circular. A *Dictionary* is an array-like structure that binds a key with a value, as a "hash table".

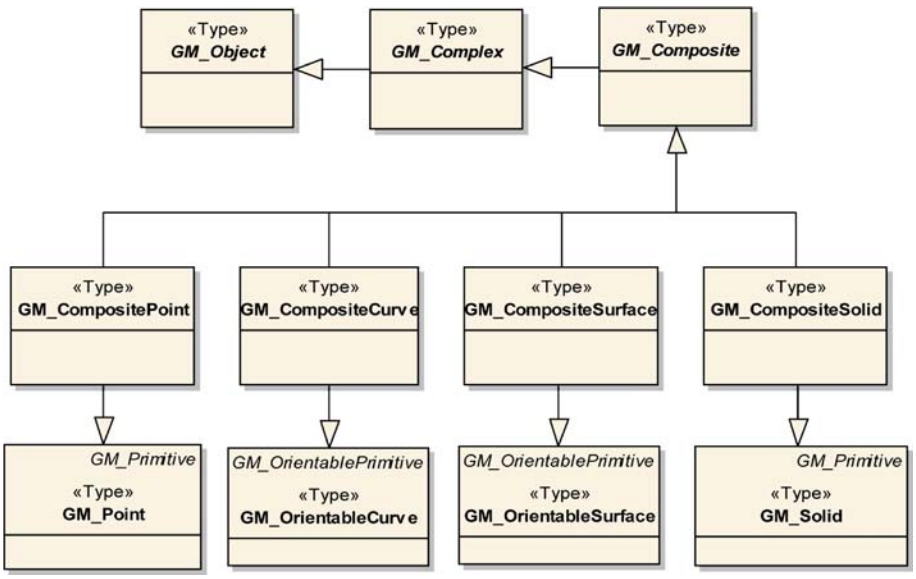
Derived types refer specifically to units of measure. A generic class UnitOfMeasure is introduced with a number of subclasses, which specify units for the different measures such as length, area, angle, time, etc.

Spatial Schema The representation of spatial characteristics is fundamental in geographic information for the description of geographic feature, either in an application schema or a geographic database. Spatial characteristics encompass the geometry of the feature, its location with respect to a coordinate reference system, and its topological properties with other features. ISO 19107 Spatial Schema [11] defines in detail the geometric and topological characteristics that are needed to describe geographic features spatially. In ISO 19107, geometric characteristics are of three types: primitive (GM_Primitive), aggregate (GM_Aggregate), and complex (GM_Complex). Figure 3 shows ISO 19107 basic geometric primitives: GM_Point, GM_Curve, GM_Surface, and GM_Solid. They provide all components needed to depict the shape and the location of simple geographic features such as buildings, towers, roads, bridges, rivers, etc.

Aggregate geometries (Fig. 4) depict features composed of multiple geometric primitives such as an archipelago,



Modeling with ISO 191xx Standards, Figure 4 Aggregate geometries specified by ISO 19107 [11]



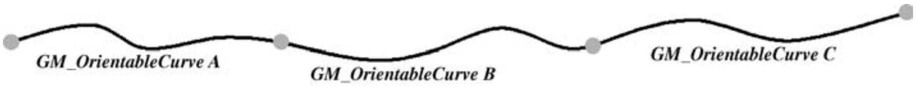
Modeling with ISO 191xx Standards, Figure 5 Complex geometries specified by ISO 19107 [11]

composed of multiple surfaces, or a campus, composed of small (or point like) and large (or surface like) buildings. Accordingly, ISO 19107 defines GM_Aggregate, which is a set of any kind of geometric types (GM_Object); GM_MultiPoint, a set of GM_Points; GM_MultiCurve, a set of GM_OrientableCurves; GM_MultiSurface, a set of GM_OrientableSurfaces; and GM_MultiSolid, a set of GM_Solids.

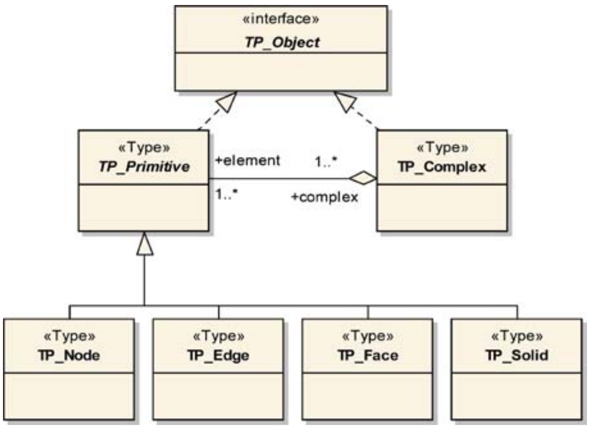
In some cases, geographic features have a more complicated geometric structure. That is the case for a road or a hydrographical network. Consequently, ISO 19107 has developed geometries for complexes: GM_Complex, GM_CompositePoint, GM_CompositeCurve, GM_CompositeSurface, GM_CompositeSolid (Fig. 5). They all consist of a set of GM_Primitives, which have disjoint interiors, i. e., the interior of one geometry does not intersect with any other geometry. In a GM_Complex, primitives are joined together only by the way of a common boundary in order to form a unique geometry. For example, a GM_CompositeCurve is made of a set of GM_OrientableCurves where the first point of each curve, except for the first, corresponds to the last point of the previous one (Fig. 6) and the final GM_CompositeCurve possesses all the properties of a GM_OrientableCurve. Similarly, a GM_CompositePoint behaves as a GM_Point, a GM_CompositeSurface as a GM_OrientableSurface, and a GM_CompositeSolid as a GM_Solid.

The location of all types of geometric primitives, aggregates, and complexes is described with coordinates that are referenced to a coordinate reference systems. This will be discussed in Sect. "Spatial Referencing".

Topological primitives are needed to support complex geometric calculations, such as adjacency, boundary, and network analysis between geometric objects. In ISO 19107, topological primitives are structured similarly to the geometric structure with the definition of topological objects (i. e., TP_Objects). Accordingly, TP_Object is found at the top level of the structure and defines operations that are inherited by all its subordinates. There are two types of TP_Objects: TP_Primitive and TP_Complex (Fig. 7).



Modeling with ISO 191xx Standards, Figure 6 Complex curve



Modeling with ISO 191xx Standards, Figure 7 Topological primitives and complex specified by ISO 19107

TP_Primitives include TP_Node, TP_Edge, TP_Face, and TP_Solid, which parallel the geometric primitives presented earlier. TP_Complex aggregates TP_Primitives of different types up to the dimension of the complex. For example, a complex referring to a road network would include TP_Nodes and TP_Edges.

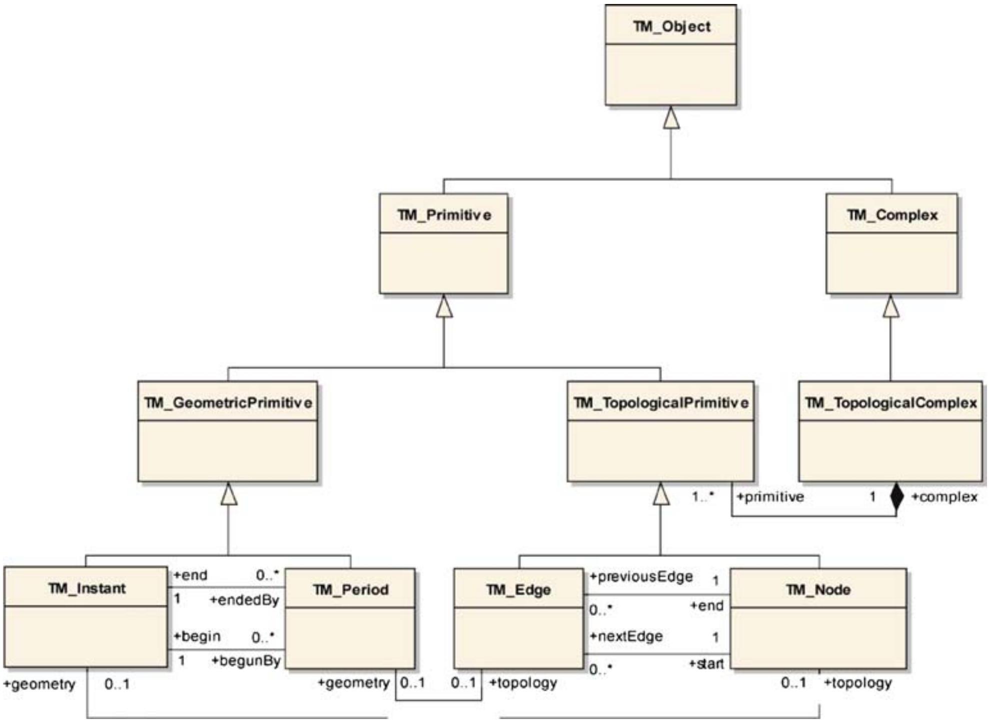
Temporal Schema The representation of geographic features or attributes in an application schema or a geographic database may also have a temporal definition. For example, a bridge could be accessible only within specific periods of time in a day. ISO 19108 Temporal Schema [12] defines primitives, topological primitives, and topological complexes for the description of temporal characteristics. Basically, there are two temporal primitives: TM_Instant and TM_Period (Fig. 8). TM_Instant is used to describe a point in the time dimension whereas TM_Period represents a temporal extent, i. e., duration. A topological primitive, TM_Node or TM_Edge, provides information about its linkage or connectivity with other topological primitives. A set of topological primitives linked together forms a temporal topological complex, i. e., a TM_TopologicalComplex.

Application Schema Geographic databases are collections of representations of geographic real world phenomena, i. e., geographic features that hold in a specific context and are of interest for users and/or applications. The manner in which a real world phenomenon is described depends on the perception of the observer about that phenomenon from which he/she abstracts characteristics of importance. Typically, an application schema is used to identify geographic as well as nongeographic features and their characteristics that are maintained in a geographic database or processed in an application.

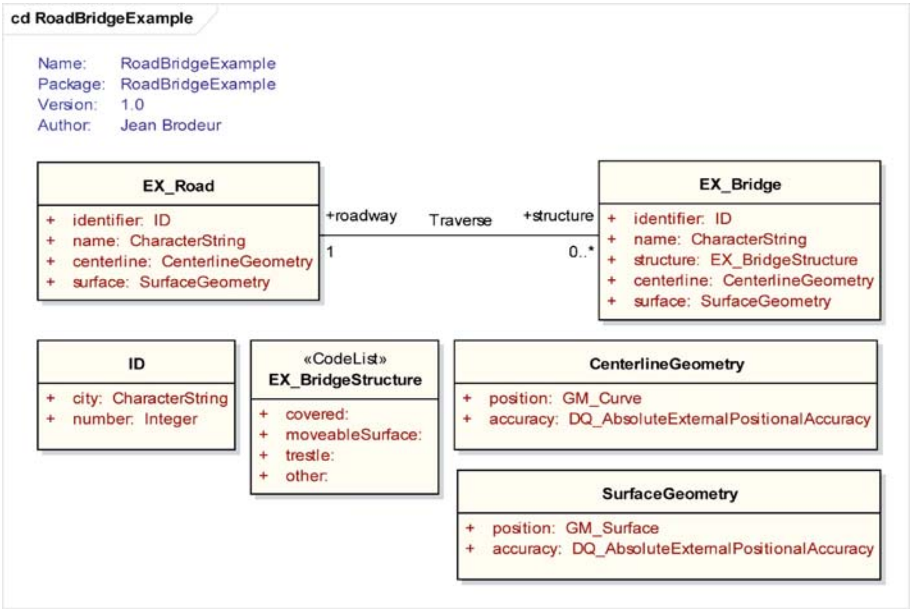
For the same universe of discourse, observers may come to different conceptual models. Consequently, it is neither desirable nor of interest to standardize conceptual models. However, ISO 19109 Rules for Application Schema defines a number of rules to develop and maintain consistent application schemas. An application schema is a conceptual schema or model for geographic data that is required by one or more applications. Consistent application schemas help in the understanding, the acquisition, the processing, the analysis, the access, the representation, and the exchange of geographic information, in other words in the interoperability. On the one hand, ISO 19109 defines a metamodel called the *General Feature Model* (GFM). The GFM includes all the necessary elements for the description of features (GF_FeatureType), their properties (GF_PropertyType, GF_AttributeType, GF_AssociationRole, and GF_Operation), the different types of associations (GF_AssociationType, GF_AggregationType, GF_SpatialAssociationType, and GF_TemporalAssociationType), generalization/specialization (GF_InheritanceRelation) relationships, and constraints (GF_Constraint). The description of an attribute encompasses thematic (GF_ThematicAttributeType), location (GF_LocationAttributeType), and temporal attributes (GF_TemporalAttributeType). On the other hand, ISO 19109 defines rules to end up with consistent application schemas. These rules explain:

- The steps to create application schemas
- The general information of an application schemas (e. g., name and version)
- The integration of application schemas
- The use of UML to build application schemas
- The inclusion of metadata in an application schemas (e. g., quality of the geometry)
- The use of geometric, topological, and temporal primitives

Figure 9 illustrates an example of an application schema complying with ISO 19109 rules. It shows two feature classes: *EX_Road* and *EX_Bridge*. *EX_Road* has an *identifier* and a *name* thematic attribute. The attribute *identifier* is of the type *ID*, which is defined in the class *ID*, and the attribute *name* of the type *CharacterString* from



Modeling with ISO 191xx Standards, Figure 8 Temporal primitives, topological primitives and complex specified by ISO 19108 [12]



Modeling with ISO 191xx Standards, Figure 9 Road-bridge application schema example

ISO 19103. *EX_Road* has also two geometric attributes: *centerline* and *surface*. The attribute *centreline* is of the type *CenterlineGeometry*, which is described in the class *CenterlineGgeometry*. *CenterlineGeometry* has an attribute *position*, which describes the geometry of the road, and an attribute *accuracy* for the description of the positional accuracy of the geometry. The class *EX_Bridge* has a similar attribute arrangement to *EX_Road* with the addition of the attribute *structure* that takes its value from the code list *EX_BridgeStructure*. Finally, *EX_Road* is linked to *EX_Bridge* through the *Traverse* association. In this association, *EX_Bridge* plays the role *structure* in *EX_Road* and, reciprocally, *EX_Road* plays the role *roadway* for *EX_Bridge*. An instance of *EX_Bridge* must always be associated to an instance of *EX_Road*.

Feature Cataloging To describe an application schema completely, it is imperative to provide the semantics of each of its elements: classes, attributes, relationships, and so on. The object of ISO 19110 Methodology for Feature Cataloging is to define a mechanism for the documentation of the semantics of all application schema elements. Basically, the methodology for cataloging features agrees to the GFM discussed in Sect. "Application Schema". A feature catalog includes a description of itself, feature types, feature property types, feature operations, feature attributes and attribute values, relationships, and association roles. As an example, the feature catalog of the road-bridge example could be depicted as in Table 1.

Spatial Referencing Typically, the representation of a geographic feature position is done according to a coordinate reference system (CRS). In ISO19111 Spatial Referencing by Coordinates [13], a CRS is defined as a coordinate system which is related to an object by a datum. For the purpose of geodesy, the object of concern is the Earth and consists in the reference for the geometric depiction of geographic features. ISO19111 provides the mechanism for describing a CRS, which can be either simple or compound. A simple CRS includes a description of the coordinate system (i. e., its axes) and the datum. A compound CRS is an aggregation of two or more simple CRSs. For example, a compound CRS may include a horizontal and a vertical CRS, which both have specific coordinate axes and datum.

CRS is of interest to conceptual modeling of geographic databases as it provides the description in which the positions of geographic features are known. Accordingly, it is important to note that each geometric object as defined in ISO 19107 (primitive, aggregate, or complex) is associated to a CRS and, as such, carries all parameters describing the CRS in which its coordinates are recorded (see Fig. 10).

A geographic feature can also be located by the way of a geographic identifier, which provides a geographic reference through a label (e. g., King West street) or a code (e. g., postal code "J1H 1P1"). Geographic identifiers are usually organized in gazetteers, which are dictionaries of geographic identifiers. ISO 19112 Spatial Referencing by Geographic Identifiers [14] specifies a mechanism and components to describe geographic references based on geographic identifiers.

In this standard, a geographic identifier is represented by a SI_LocationInstance data type, which can be used in an application schema for documenting the location of

Modeling with ISO 191xx Standards, Table 1 Road-bridge feature catalog example

FC_FeatureCatalogue
- Name: RoadBridgeExample
- Scope: Transportation network
- Field of application: Tracking and routing
- Version number: 1.0
- Producer:
  - Name: Jean Brodeur
  - Role: Custodian
- Functional language: English (ISO639_2.eng)
- Feature type: EX_Road, EX_Bridge

FC_FeatureType
- Name: EX_Bridge
- Definition: Structure erected along a travelled route to span a depression or obstacle and ensure the continuity of the road and railway network
- Code: 2139
- Abstract: False
- Feature catalogue: RoadBridgeExample
- Chracteristics: identifier, name, structure, centerline, surface

FC_FeatureAttribute
- Name: identifier
- Definition: unique identifier of the object
- Cardinality: 1
- Code: 2139.2
- ValueType: ID

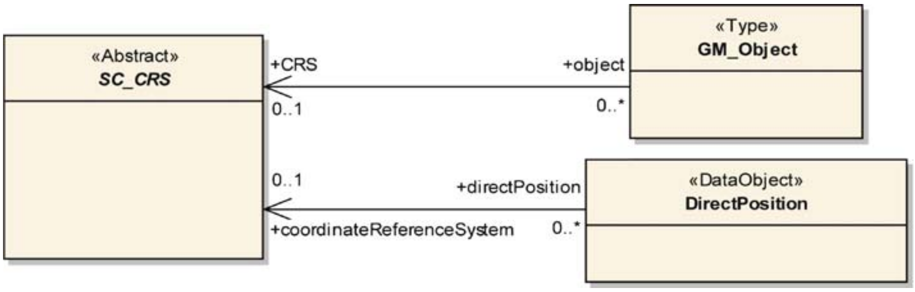
FC_FeatureAttribute
- Name: name
- Definition: place-name of the feature
- Cardinality: 1
- Code: 2139.2
- ValueType: CharacterString

FC_FeatureAttribute
- Name: structure
- Definition: kind of construction
- Cardinality: 1
- Code: 2139.3
- ListedValue: covered, moveable surface, trestle, other

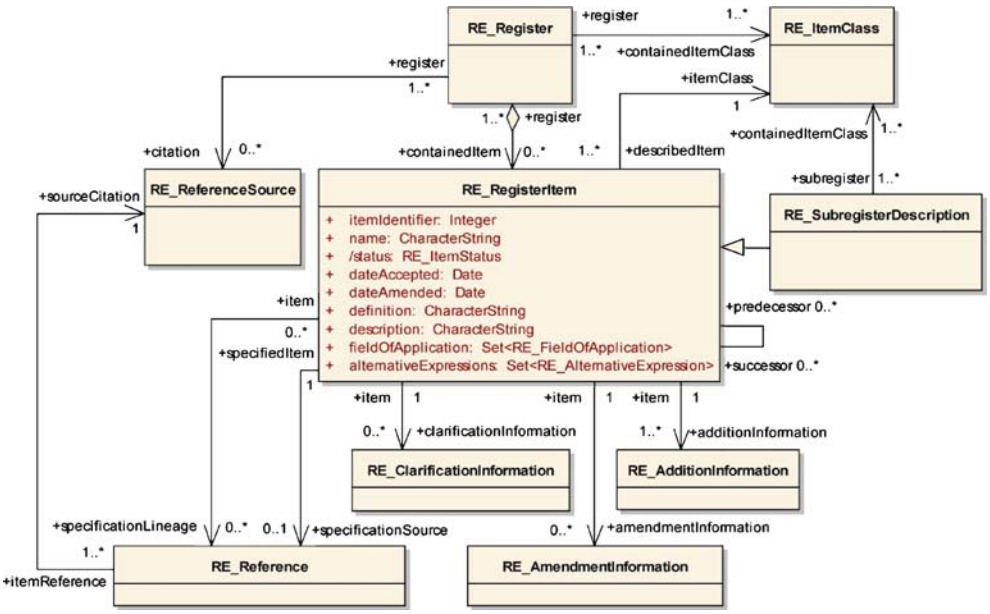
FC_ListedValue
- Label: covered
- Code: 1
- Definition: A bridge that has a building like cover to protect the bridge deck

FC_ListedValue
- Label: moveable surface
- Code: 2
- Definition: A bridge of which section can be moved to allow passage of vessels . . .

M



Modeling with ISO 191xx Standards, Figure 10 Geometric object relationships with coordinate reference systems (*CRSs*)

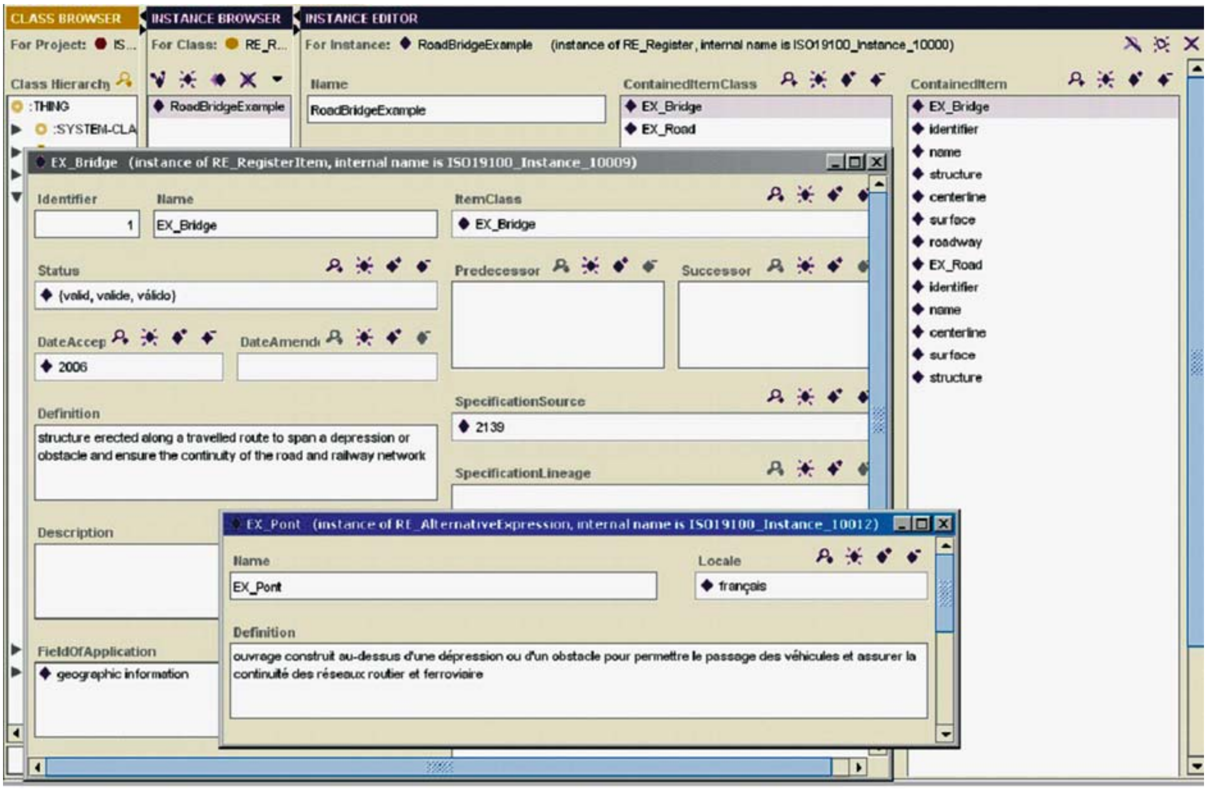


Modeling with ISO 191xx Standards, Figure 11 Geometric object relationships with CRSs

a geographic feature. For example, "J1H 1P1" constitutes a SI_LocationInstance from the gazetteer of postal code of Canada and locates the north side of a portion of the *King West street* in *MyCity*.

Metadata Metadata, commonly known as data about data, is meaningful information to better understand geographic data elements and data sets. ISO 19115 Metadata [8] sets the content and structure of geographic metadata. It covers topics like identification, constraints, quality, lineage, maintenance, spatial representation, reference system, content, portrayal, distribution, and application schema. Metadata elements are used to describe data sets and to provide additional information about geographic feature characteristics. For example, the road-bridge application schema illustrated in Sect. "Application Schema" uses an attribute accuracy of DQ_AbsoluteExternalPositionAccuracy type from ISO 19115 in classes CenterlineGeometry and SurfaceGeometry to report the geometric accuracy of road and bridge instances. This data type and others from ISO 19115 can be used to report metadata about different facets of geographic data and therefore to benefit conceptual modeling. In addition, ISO 19115 includes the application schema (MD_ApplicationSchemaInformation) and a reference to the feature catalog (MD_ContentInformation) as metadata items for the description of datasets, data collections, or series. This makes the application schema and the feature catalog living components for the use of geographic information to support better understanding of the data but also analysis and reasoning purposes. Consequently, metadata contributes to semantic interoperability of geographic information.

Register According to ISO 19135 Procedures for Item Registration [15], a register consists of a collection of object identifiers with definitions recorded in a file or set of files. It establishes the identity of concepts of interest with-



Modeling with ISO 191xx Standards, Figure 12 Road-bridge register example

in a namespace for geographic applications and databases. A registry is a complete system to ensure the appropriate management of a register including a register owner, a register manager, and submitting organizations, which sponsor the register.

ISO 19135 establishes the structure of a register (RE_Register) and the rules for the proper management of registers. A register contains items that exist as components of an item Class (Fig. 11). An item (RE_RegisterItem) is described by an identifier, a name, a status, an acceptance date, an amendment date, a definition, a description, a field of application, and a set of alternative expressions. Items may evolve in time and a register allows for maintainance and tracking of modifications.

Registers contribute to geographic database modeling in maintaining semantics of concepts that application schemas, feature catalogs, or geographic databases may reuse and associate to model elements or data. They support concepts' multilingual representation via alternative expressions that are associated with a specific locale (i. e., the language and country identification). Figure 12 illustrates a register of the road-bridge example that shows EX_Bridge as a contained item of the register with EX_Pont as an alternative expression in French.

Key Applications

The importance of spatial database modeling based on the ISO191xx suite of standards lies mostly in increasing interoperability of geographic information. Geographic data described in a common structure using the same data types facilitates their access, understanding, integration, and use through SDIs. It is easier for users to find and get geographic data that fit their application when, for example, they are published on a web portal such as the CGDI Discovery portal in Canada (http://geodiscover.cgdi.ca/gdp/) or Geospatial One Stop in the US (http://gos2.geodata.gov/wps/portal/gos) using these standards where all the information is described and presented similarly. From a user point of view, it supports applications such as disaster management, global warming, sustainable development, traffic management, etc. which typically need to integrate data from different sources, themes, levels of details, such as road, drainage, railways, satellite images, relief and so on.

Future Directions

The contribution of standard-based geographic data modeling to interoperability of geographic information and the contribution of international standards in geographic information to geographic data modeling has been explained in this chapter.

Standards in geographic information play an important role in geographic data modeling. As noted here, a number of standards provide data types for the representation of geographic information. These data types include basic types, such as numbers, texts, and dates, geometric types (e. g., point, line and surface), temporal types (e. g., instant and period), etc. This is the case for ISO/TS 19103 (conceptual schema language), ISO 19107 (spatial schema), and ISO 19108 (temporal schema). Other standards provide data types for the description of ancillary information. This is the case for ISO 19111 (spatial referencing by coordinates), ISO 19112 (spatial referencing by geographic identifiers), ISO 19115 (metadata), and ISO 19131 (data product specifications). Finally, there are standards that concern the elaboration of geographic data models. These standards provide structures and rules for the elaboration of application schemas and documentation of concepts. This is the case for ISO 19109 (rules for application schema), ISO 19110 (methodology for feature cataloging), and ISO 19135 (procedures for item registration).

Additionally, standards in geographic information bring a common domain ontology, which describes a set of concepts that are needed for the overall aspects of geographic information. This sets the foundation for the Semantic Web in the geographic information realm. Some open frameworks like GeOxygene [16] are already available to help in the interoperable development and deployment of geospatial applications over the internet.

Cross References

▶ Geospatial Semantic Web, Interoperability
▶ Metadata and Interoperability, Geospatial
▶ Modeling with Pictogrammic Languages
▶ OGC's Open Standards for Geospatial Interoperability
▶ Vector Data

Recommended Reading

1. ISO/TC211: ISO19109:2005 Geographic Information—Rules for Application Schema. ISO, Geneva, Switzerland (2005)
2. Brodeur, J., Bédard, Y., Proulx, M.J.: Modeling geospatial application databases using UML-based repositories aligned with international standards in geomatics. In: Proceedings of the 8th ACM Symposium on Advances in Geographic Information Systems (ACMGIS), Washington, D.C., USA, 6–11 Nov 2000
3. ISO/TC211: ISO19110:2005 Geographic Information—Methodology for Feature Cataloguing. ISO, Geneva, Switzerland (2005)
4. ISO/TC211 (2007) Geographic information/geomatics homepage. http://www.isotc211.org/. Accessed 2 Sept 2007
5. OGC (2007) Open Geospatial Consortium homepage. http://www.opengeospatial.org/. Accessed 2 Sept 2007
6. Institute of Electrical and Electronics Engineers: IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. IEEE Computer Society, USA (1990)
7. Brodeur, J., Bédard, Y., Edwards, G., Moulin, B.: Revisiting the concept of geospatial data interoperability within the scope of a human communication process. Trans. GIS **7**, 243–265 (2003)
8. ISO/TC211: ISO19115:2003 Geographic Information—Metadata. ISO, Geneva, Switzerland (2003)
9. ISO/TC211: ISO/TS19103:2005 Geographic Information—Conceptual Schema Language. ISO, Geneva, Switzerland (2005)
10. Object Management Group: Unified Modeling Language: Superstructure, v2.0. OMG, Needham, MA, USA (2005)
11. ISO/TC211: ISO19107:2003 Geographic Information—Spatial Schema. ISO, Geneva, Switzerland (2003)
12. ISO/TC211: ISO19108:2002 Geographic Information—Temporal Schema. ISO, Geneva, Switzerland (2002)
13. ISO/TC211: ISO19111:2002 Geographic Information—Spatial Referencing by Coordinates. ISO, Geneva, Switzerland (2002)
14. ISO/TC211: ISO19112:2003 Geographic Information—Spatial Referencing by Geographic Identifier. ISO, Geneva, Switzerland (2003)
15. ISO/TC211: ISO19135 Geographic Information—Procedures for Item Registration. ISO, Geneva, Switzerland (2005)
16. Badard, T., Braun, A.: OXYGENE: A platform for the development of interoperable geographic applications and web services. In: Proceedings of the 15th International Workshop on Database and Expert Systems Applications (DEXA'04), Zaragoza, Spain, 30 Aug–3 Sept 2004

Modeling with Pictogrammic Languages

Yvan Bédard, Suzie Larrivée
Dept of Geomatics Sciences, Centre for Research in Geomatics, Canada NSERC Industrial Research Chair in Geospatial Databases for Decision Support, Laval University, Quebec City, QC, Canada

Synonyms

Spatial modeling language extension; Spatio-temporal modeling language extension; Perceptory pictograms

Definition

"Spatial databases" consist of large groups of data structured in a way to represent the geographic features of interest to the users of a system. Spatial database models are schematic representations of these data. Database models are created to design and document the system, to facilitate communication and to support programming. They are created using CASE tools (computer-assisted software engineering). CASE tools support schema drawing, dictionaries and code generation. Database schemas are typically represented with a graphical language such as

UML (Unified Modeling Language; see http://www.uml.org and [11]).

"Database models" can represent (1) users' real-life views of the data of interest, (2) developers' views of the potential organization of these data for a family of technologies, or (3) their final implementation on a specific platform. For example, in the standard Model-Driven Architecture (MDA) method (http://www.omg.org/mda/), these three models represent three levels of abstraction and are respectively called CIM (computation-independent model), PIM (platform-independent model) and PSM (platform-specific model). In other methods, they may be called conceptual, logical and physical models as well as analysis, design and implementation models.

"Pictograms" are symbols aimed at facilitating modeling. Different sets of pictograms have been proposed. This chapter presents those used by the CASE tool Perceptory (http://sirs.scg.ulaval.ca/perceptory) since they are the most widely used, they were designed to allow developers to keep their method, and they were thoroughly tested as implementations of UML stereotypes. In Perceptory, they aim at hiding the complexity of geometric primitives in CIM and PIM models. They can serve other purposes as well and have been implemented in other CASE tools [14].

Historical Background

In the field of GIS, pictograms were first proposed in 1989 by Bedard and Paquette [6] to simplify how Entity-Relationship (E/R) models depicted the geometry of cartographic features. It was then called "Sub-Model Substitution" technique as the main goal was to remove from the spatial database model those geometric primitives with their data elements and relationship (considered of no interest to the user) and to replace them by simple symbols showing only the information of interest to the users (i. e. the features' shape). This first solution was tested in several projects and enhanced over time to lead to the development of Modul-R [4,5,7], the first spatio-temporally extended E/R which led to Orion, the first GIS-compatible CASE tool in 1992 [5]. This first solution has influenced several researchers afterwards. Examples of methods or tools using pictograms for spatial databases include Perceptory [1,3] which is used in over 30 countries, Software Development Process Model with Objecteering [14], MADS [17], CONGOO [16], UML-Geoframe with Argo-CASEGEO [10], and STER [19].

In 1996, Modul-R pictograms were revisited to integrate three paradigms: object-orientation (OO), plug-in (module, blade, cartridge) and a pragmatic symbiotic approach [3]. Object-orientation allowed for more expressive power and was first tested with UML in its pre-release days. The plug-in approach led to define the pictograms and their syntax as a module, i. e. a specialized language designed to extend standard languages (e. g. UML, E/R, English). This allowed for enriching one's modeling language and tool rather than requiring to adopt new ones. For instance, in addition to Perceptory, these pictograms have been used with commercial and open-source CASE tools such as Oracle Designer, Objecteering and others while being also used to describe spatial integrity constraints, to compare database semantics and to improve software user-interfaces. With regards to the symbiotic approach, it came from cognitive studies and pragmatics lessons resulting from several projects with practitioners, including very complex ones. It helped to find a better balance between human abilities, language requirements, database design methods and commercial software constraints. Practical projects clearly indicated the need to better support unexpected complex situations, to simplify the pictograms along with their syntax, and to better balance the content of the graphical schema with the ontological content of the dictionary (i. e. simpler schemas, increased use of natural and formal languages in the dictionary). This was a departure from the trend of that period to rely increasingly on graphical depictions. Such novel approach and the arrival of UML led to developing Perceptory. This approach also goes beyond the leading tendency to perceive "modeling" solely as a schema-building exercise since in fact it is not; a schema without clear and complete semantics is meaningless and its robustness cannot be validated. Accordingly, good spatial database modeling becomes an ontological exercise. For example, Perceptory provides specialized spatial and temporal sections in its dictionary (as can be added to other CASE tools). In the remaining of this chapter are presented the scientific fundamentals of modeling spatial databases with pictograms, using examples from the UML-based Perceptory CASE tool.

Scientific Fundamentals

"Pictograms" aim at supporting the expression of any feature's spatial and spatio-temporal properties into a consistent manner that is compatible with various human-oriented languages (ex. UML, Entity-Relationship, English, French).

"Syntax rules" dictate the way to combine and position pictograms in a model or document. These rules also dictate how to use special characters (0–9 N ,). Properly combining pictograms, with or without characters, makes it possible to express complex cases of geometry and spatio-temporality, namely: facultative, mandatory, alternate, aggregate, multiple, and derived.

M

Modeling with Pictogrammic Languages, Table 1 Simple pictogrammic expression for geometry

	2D space	3D space	Examples of cases
0D geometry			hydrants when they are all represented by points
1D geometry			road segments when they are all represented by lines
			electric poles when they are all represented by vertical lines
2D geometry			lakes when they are all represented by polygons
			walls when they are all represented by vertical plans
3D geometry			buildings when they are all represented by solids

Modeling with Pictogrammic Languages, Table 2 Simple pictogrammic expressions for temporality

		Examples of cases
0D temporality		Existence of accidents; traffic flow of a road segment
1D temporality		Existence of a building; duration of its commercial use; duration of its ownership by a given person

A "pictogrammic expression" includes one or several pictograms which are positioned in a precise manner with pertinent digits according to a syntax. Such a pictogrammic expression completely describes the spatial, temporal or spatio-temporal properties of either (1) a feature, (2) where and when an attribute value is valid within an object geometry or existence, or of (3) a relationship between features. For example, in Perceptory, the simple expression is made of only one pictogram and represents a simple 1D geometry in a 2D universe. Similarly, the expression represents the same geometry in a 3D universe while the expression adds thickness to this geometry. On the other hand, the expression (i. e. 1D OR 2D) has a different meaning from the previous ones and from the expression (i. e. 1D + 2D) or from the expression 0,N. In a similar manner, the simple expression represents one instant, the expression represents one period of time. More complex temporal and spatio-temporal expressions can be made.

Grouping pictograms and syntactic rules commonly used together allows one to form a specialized graphical language called "PVL" (Plug-in for Visual Languages). A PVL, as introduced in [3], allows extending a modeling language with a tested method that is compatible with other PVLs if needed. For example, one may decide to use only a small group of Perceptory pictograms to make a 2D spatial PVL (i. e. a language to depict plane geometries of geographic features) while later on, if needed, use additional Perceptory pictograms to have a 3D spatio-temporal PVL. A pictogrammic expression is sometimes called a PVL expression.

The pictograms high level of abstraction facilitates the making of database models, reports, specifications, spatio-

id	gravity	nb_death	nb_injuries	nb_vehicles	date	x_geom	y_geom
1	2	0	0	2	07/07/2005	-71,2	46,78
2	2	0	1	3	03/04/2006	-71,35	45,9
3	1	3	2	2	04/05/2006	-71,56	45,64

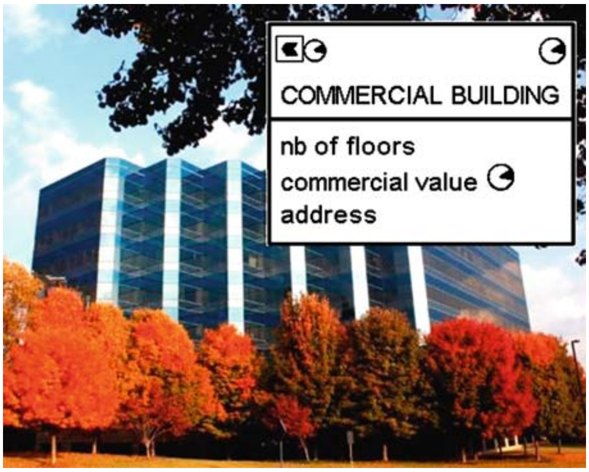
Modeling with Pictogrammic Languages, Figure 1 Example of simple pictogrammic expressions for the geometry and existence of a UML object class Accident temporal integrity constraints, user interfaces, and similar tasks of a system development workflow. They hide the complexity inherent to the description of geometric and temporal primitives and relationships as well as implementation and standard-related issues. In particular, they facilitate the building, editing, communication and validation of spatio-temporal database models as well as their translation into efficient data structures. In spite of such translation rules, the PVL are independent from commercial software and numerous standards.

The pictograms were first created for spatial database modeling and are best described in such a context. Accordingly, the present chapter describes the pictograms implemented as UML stereotypes in Perceptory object class model. In such a context, the PVL allows the analyst or designer to describe the spatial and temporal properties of the elements depicted in an object class schema. Perceptory pictograms support 0D, 1D, 2D and 3D geometries for objects located in 2D or 3D universes (see Table 1). Supported temporalities are 0D (instant) and 1D (period) (see Table 2). Supported combinations are simple, complex (aggregate), alternate (exclusive OR), multiple (AND), spatio-temporal and hybrid (combinations of any of the above) (see Tables 3, 4 and 5). Supported minimum multiplicities include facultative (0), mandatory (1), specific number, and many (N), while maximum multiplicities include the three latter. Special cases are "any possibility", "not yet defined" and "complicated", the latter pointing to a textual description in the repository (when easier to read). All geometries and temporalities can be indicated

Modeling with Pictogrammic Languages, Table 3 Syntax for advanced 2D and 3D spatial pictogrammic expressions

Geometry	Examples of syntax	Examples of cases
Aggregate geometry		
(complex)		Hydrographic networks composed of 1D rivers and 2D lakes (i. e. aggregate of different geometries)
(simple)	1,N	Some municipalities may include several 2D geometries such as islands (i. e. aggregate of similar geometries)
Alternate geometry (on same line)		Buildings having a 0D shape if area < 1 hectare OR a 2D shape if area > 1 hectare (Exclusive OR)
Facultative geometry	0,1	Buildings in database may have no geometry if area < 0.2 hectare, or a 0D shape if area > 0.2 hectare
Multiple geometry (on different lines)		Every municipality has a 2D shape AND a 0D location (ex. downtown). See [1] for detailed examples.

N.B. same syntax for 2D and 3D pictograms



Modeling with Pictogrammic Languages, Figure 2 Example of a spatio temporal pictogrammic expression, a temporal expression for the existence of the UML object class and of another one to keep track of the evolution of one attribute

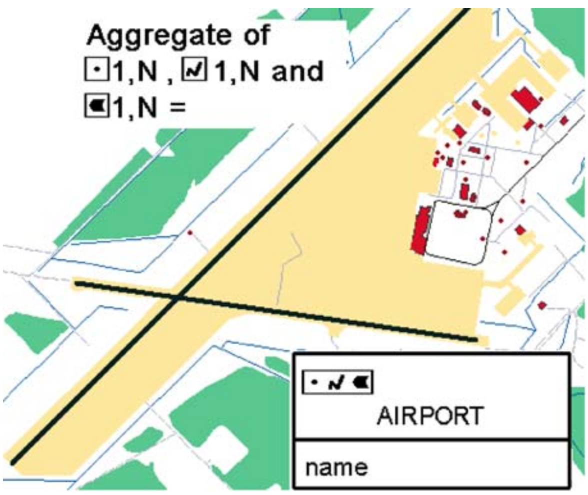
Modeling with Pictogrammic Languages, Table 4 Syntax for advanced temporal and spatio-temporal pictogrammic expressions

Temporality	Examples of syntax	Examples of cases for feature existence and states
Alternate temporality (on same line)		Forest fires lasting several days OR 1 day (if temporal resolution is 1 day); water level data varying continuously when opening/closing the dam OR remaining stable for a period once a level is reached
Facultative temporality	0,1	Houses in database may need NO construction and demolition dates IF area < 0.2 hectare
Multiple temporality (on different lines)		Hurricane existence defined by a date of beginning and a duration for some purposes, AND by a unique date of maximum peek for other purposes. Buildings commercial value considered stable for the whole year for tax purposes but as being valid only the day when the building was assessed for market analysis purposes.
Spatio-temporality		Position of a moving vehicle. The temporal pictogram affects the spatial pictogram on its left

N.B. Selecting between or depends on the temporal granularity defined into the repository for each class, attribute and geometry.

Modeling with Pictogrammic Languages, Table 5 Syntax and pictograms for special cases

Derived geometry or temporality "italic pictogram"		Municipality centroids derived from their polygons; 3D buildings derived from 2D buildings with number of floors; duration of commercial use derived from permits
Hybrid expression (combination of any pictos above)	1,N	A set of individual cyclists continuously moving during a race or forming a group that changes its size during the race
Default multiplicity	If no multiplicity is written immediately after a pictogram, the 1,1 multiplicity is implied	
Any possibility		"wildcard pictogram" meaning no predefined shape or temporality, and no restriction on the geometry or temporality
Complicated		Better explained textually in the dictionary than using a complicated PVL expression in a schema. Replaces a long hybrid expression if desired.
Not yet defined		During the process of designing a database, one may anticipate a need for geometry or temporality, but ignore which one and will replace it later by a regular pictogram



Modeling with Pictogrammic Languages, Figure 3 Example of a complex aggregate geometry for Airport object class, that is an aggregate of points, lines ans polygons (Data from ministère des Ressources naturelles et de la faune du Québec)

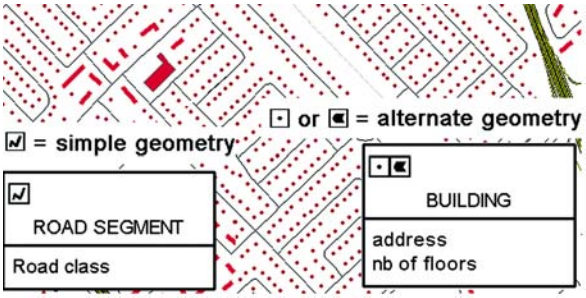
as "measured" or "derived from other attributes, objects, relationship using calculations, spatial or temporal analysis". Having no geometry or temporality is also accepted. Pictogrammic expressions may describe object classes, association classes, attributes, and may be used within operations.

Examples of the use of pictogrammic expressions for UML object classes are presented hereafter. Figure 1 describes an accident as a an instantaneous event with a geometry defined as a point. Figure 2 shows a case where users want to keep information about the existence of commercial buildings (dates of construction and destruction), about the evolution of their commercial value (attribute data with their period of validity) and about the evolution of its polygonal representation if it is enlarged or modified. Figure 3 illustrates a case of aggregated complex geometry while Fig. 4 shows cases of simple and of alternate geometries. At last, Fig. 5 shows a case of multiple geometry where the first pictogram expresses the fact that every building is represented by simple polygon at large scales and the second line of pictograms indicates that some (but not all) buildings may have a second geometry, either a point or a line, depending on their size, for small scale maps (usually to properly place symbolic representations). See [2] for more details.

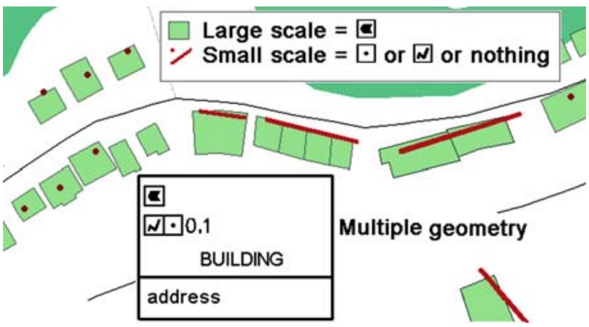
From a UML point of view, these pictogrammic expressions are implemented as stereotypes (a formal way of extending UML) and are built on-the-fly in Perceptory. Using such pictogrammic expressions has also proved to be useful to model spatial multidimensional databases (or datacubes) as used in spatial data warehousing and SOLAP (Spatial On-Line Analytical Processing). These datacubes pictogrammic expressions include datacube, data dimension, member, measure and are compatible with the previous spatial and temporal pictograms. They are all supported by Perceptory.

Key Applications

Pictogrammic languages, if sufficiently expressive and usable, can serve several purposes. The following paragraphs further describe the primary key application, i. e. spatial database modeling, plus other applications of interest.



Modeling with Pictogrammic Languages, Figure 4 Example of a single geometry pictogrammic expression (where each instance is represented by one line) and of an alternate geometry (where small buildings are represented by a point and large ones by a polygon) (Data from ministère des Ressources naturelles et de la faune du Québec)



Modeling with Pictogrammic Languages, Figure 5 Example of a multiple geometry pictogrammic expression, where at large scale (ex. 1:1000), buildings are represented by a polygon and at small scale (ex. 1:20 000), they are represented by a point, a line or nothing (Data from Research and Development Defence Canada and from ministère des Ressources naturelles et de la faune du Québec)

Using Pictogrammic Languages for Spatio-temporal Database Modeling

Modeling databases for GIS applications has always posed several challenges for system analysts, system developers as well as for their clients whose involvement into the development of such a project is not a familiar endeavor. Used with well-known modeling techniques, pictogrammic expressions help to meet these challenges [1,10,16,17] and are commonly used in different methods such as relational database design with UML (cf. the UML relational stereotypes in [15]). Extending CASE tools and modeling methods in such a way allows analysts and designers to work at a higher level of abstraction for the first steps of a spatial database project. As presented in Fig. 6, once high-level models are completed (e. g. Perceptory CIM), they can be translated and enriched to give more technical models which are closer to implementation (e. g. OGC PIM, models by Brodeur and Badard in the chapter entitled "Modeling with ISO 191xx Standards") and finally highly technical models which specific to one implementation on one platform (e. g. ESRI Shapefiles PSM). Such multi-level approach is typical of good software engineering methods as exemplified in Fig. 6 and by the following landmark methods:

- The Object Management Group (OMG) Model-Driven Architecture (MDA) which has three levels of models: Computation Independent Model (CIM), Platform Independent Model (PIM) and Platform Specific Model (PSM);
- Zachman Framework [12,18] which has a business or enterprise model, a system model and a technology model (also called semantic, logical and physical models);
- Rational Unified Process (RUP) which has a domain model, an analysis model, a design model and an implementation model.

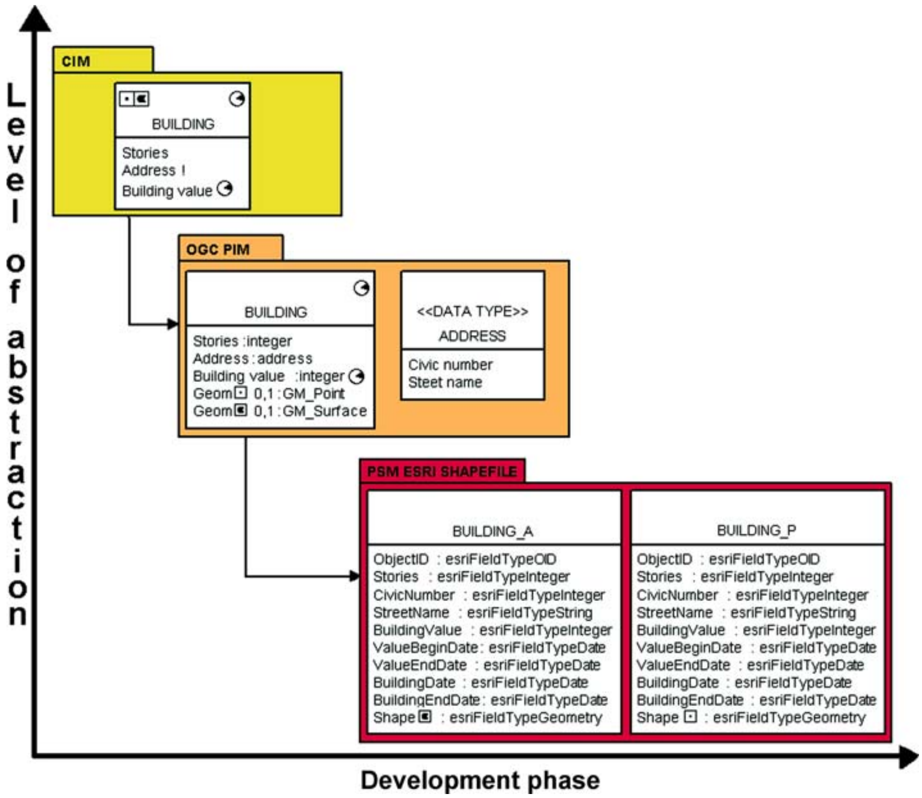
Since the pictograms are aimed at facilitating modeling by being closer to human language than typical modeling artefacts, they are primarily used in high-level models. Regarding the MDA method, pictogrammic expressions are more widely used for CIM than for PIM and PSM:

- CIM: "A *computation independent model* is a view of a system from the computation independent viewpoint. A CIM does not show details of the structure of systems. A CIM is sometimes called a domain model and a vocabulary that is familiar to the practitioners of the domain in question is used in its specification."[13]
- PIM: "A *platform independent model* is a view of a system from the platform independent viewpoint. A PIM exhibits a specified degree of platform independence so as to be suitable for use with a number of different platforms of similar type." [13]
- PSM: "A *platform specific model* is a view of a system from the platform specific viewpoint. A PSM combines the specifications in the PIM with the details that specify how that system uses a particular type of platform." [13]

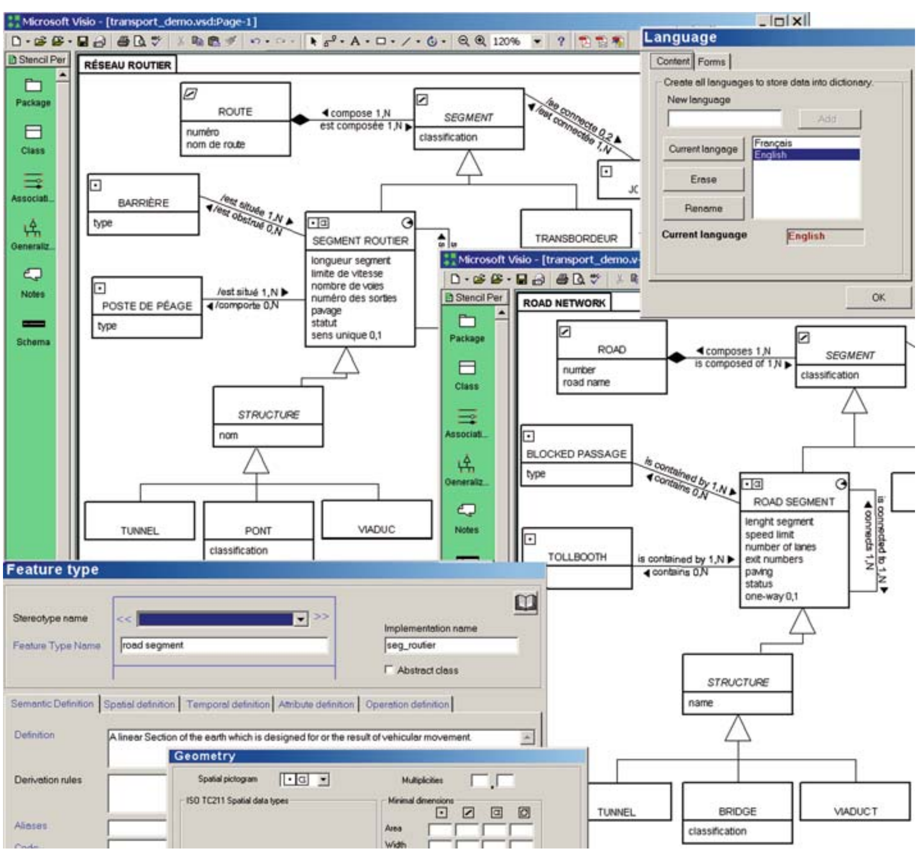
Furthermore, since pictograms are not tied to a specific natural language, they facilitate the translation of database models. For example, in Canada, several schemas and repositories are available in English and French. Figure 7 shows such French and English schemas that are synchronized thru the same repository and pictograms. The use of formal ISO-19110 labels (in blue) further facilitates communication while the use of pictograms facilitates automatic GIS code generation and bilingual reporting.

At the CIM level, pictogrammic expressions are intuitive and independent of domain ontologies and technology-oriented standards. No technology artefacts nor standardization elements must appear unless they are useful and intuitive. When the CIM is well defined, it can be translated

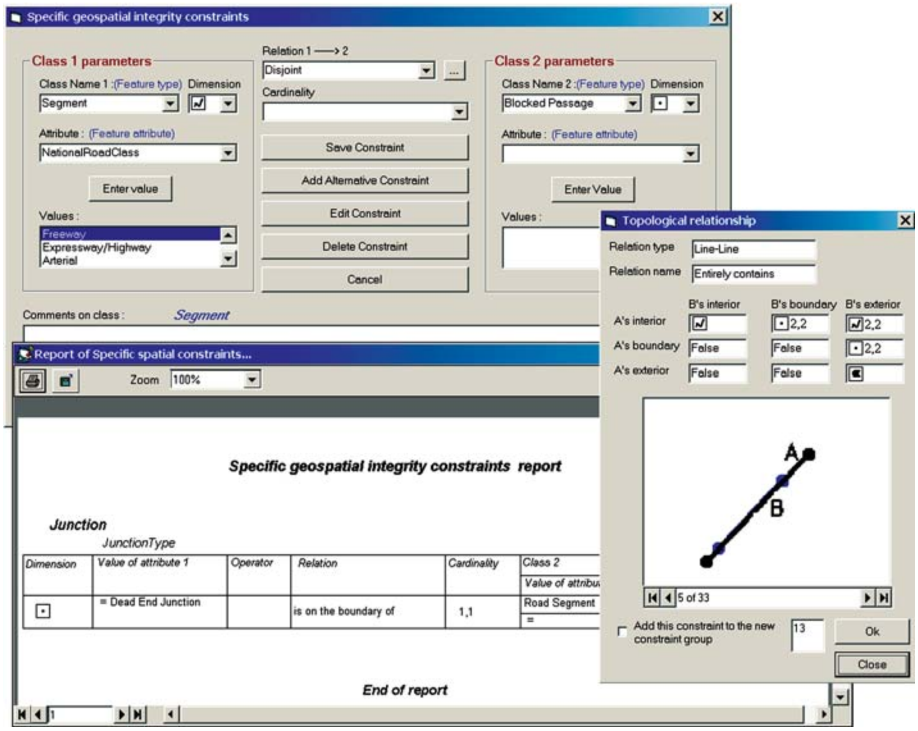
M



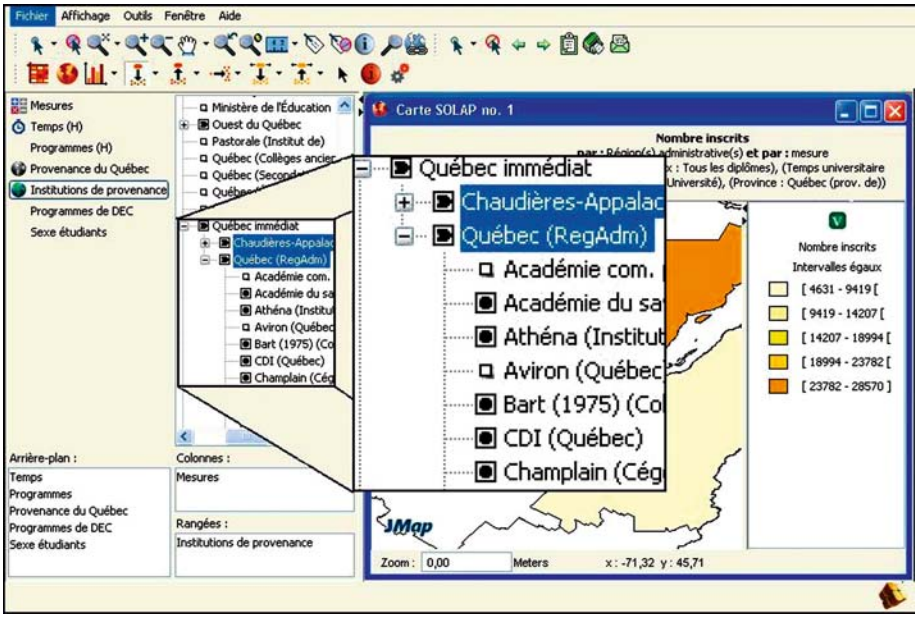
Modeling with Pictogrammic Languages, Figure 6 Examples of CIM, PIM and PSM levels of abstraction of the MDA method for a same application, where the information encapsulated in the higher levels using pictograms is expanded in the lower levels



Modeling with Pictogrammic Languages, Figure 7 Example of common pictograms in a French and an English CIM synchronized for a same spatio-temporal database using Perceptory multi-standard and multi-language capabilities



Modeling with Pictogrammic Languages, Figure 8 Examples of pictogrammic expressions to define topological constraints between two object classes (upper left), to print them in a report (lower left) and to describe them in an extended ISO e-related 3 × 3 matrix



Modeling with Pictogrammic Languages, Figure 9 JMap SOLAP interface using pictogrammic expressions

and enriched to produce lower-level models semi-automatically. Then, technology-oriented artefacts and standard-based elements replace the pictogrammic expressions. For example, in Fig. 6, the CIM evolves in a PIM where the geometry is expressed according to ISO/OGC. Then, the PSM shows the structure of two shapefiles needed to implement Building Points and Building Areas.

In addition to hiding the technical complexities of GIS and Universal server database engines, using pictogrammic expressions also hides the intricacies of international standards such as ISO/TC-211 and OGC. For example, ISO jargon doesn't express directly all possible geometries (ex. alternate and facultative geometries) and they are not cognitively compatible with clients' conceptual view who assumes a topologically consistent world (ex. GMPoint vs TPNode, GMCurve vs TPEdge, GMSurface vs TPFace, Aggregate vs Multi).

Using Pictogrammic Expressions to Define Spatial Integrity Constraints

Spatial integrity constraints can also be defined efficiently with pictogrammic expressions. For example, in Fig. 8, the upper window shows a user interface for the definition of spatial integrity constraints between two object classes, with or without considerations to specific attribute values. The lower window shows a report showing the defined spatial integrity constraints. The last window shows an example of using pictogrammic expressions in a 3×3 e-relate matrix.

Additional Usages of Pictogrammic Expressions: Software User Interfaces, Reports and Semantic Proximity Analysis

Pictogrammic expressions are regularly used in a text to express the spatiality and temporality of objects. They have been used in reports, data dictionaries and data acquisition specifications. They were also used for semantic proximity analysis [9] and integrated in a commercial package (JMap SOLAP, Fig. 9).

Future Directions

Over the last two decades, different pictogrammic languages have emerged to improve the efficiency of systems analysts and to improve the quality of spatial database design. The language presented in this chapter was the first such language and has become the most widely used one, not only within Perceptory but also in other CASE tools and in diverse applications. It uses a downloadable font (http://sirs.scg.ulaval.ca/YvanBedard/english/others.asp). Such languages will likely evolve in two major directions. First, they will accommodate the most recent spatial database trends, that is spatial datacube structures for data warehousing and SOLAP applications. Second, as they can be translated into ISO and OGC primitives [8], official adoption of such a language should be put forward to improve interoperability between spatial application database schemas, between ontologies and between other documents.

Cross References

► Modeling and Multiple Perceptions
► Modeling with Enriched Model Driven Architecture
► Modeling with a UML Profile
► Modeling with ISO 191xx Standards
► Spatio-temporal Database Modeling with an Extended Entity-Relationship Model

Recommended Reading

1. Bédard, Y., Larrivée, S., Proulx, M.-J., Nadeau, M.: Modeling Geospatial Databases with Plug-Ins for Visual Languages: A Pragmatic Approach and the Impacts of 16 Years of Research and Experimentations on Perceptory. In: Wang, S. et al. (eds.): Conceptual Modeling for Advanced Application Domains. Lecture Notes in Computer Science, Vol. 3289, pp. 17–30. Springer-Verlag, Berlin/Heidelberg (2004)
2. Bédard, Y., Proulx, M.-J., Larrivée, S., Bernier, E.: Modeling Multiple Representations into Spatial Data Warehouses: A UML-based Approach. ISPRS WG IV/3, Ottawa, Canada, July 8th–12th, p. 7 (2002)
3. Bédard, Y.: Visual Modeling of Spatial Database, Towards Spatial PVL and UML. Geomatica **53**(2), 169–185 (1999)
4. Bédard, Y., Caron, C., Maamar, Z., Moulin, B., Vallière, D.: Adapting Data Model for the Design of Spatio-Temporal Database. Comput., Environ. Urban Syst. **20**(l), 19–41 (1996)
5. Bédard, Y., Pageau, J., Caron, C.: Spatial Data Modeling: The Modul-R Formalism and CASE Technology. ISPRS Symposium, Washington DC, USA, August 1st–14th (1992)
6. Bédard, Y., Paquette, F.: Extending Entity/Relationship Formalism for Spatial Information Systems. AUTO-CARTO 9, 9th International Symposium on Automated Cartography, ASPRS-ACSM, Baltimore, USA, pp. 818–827, April 2nd–7th (1989)
7. Bédard, Y., Larrivée, S.: Développement des systèmes d'information à référence spatiale: vers l'utilisation d'ateliers de génie logiciel. CISM Journal ACSGC, Canadian Institute of Geomatics, Canada, **46**(4), 423–433 (1992)
8. Brodeur, J., Bédard, Y., Proulx, M.-J.: Modeling Geospatial Application Database using UML-Based Repositories Aligned with International Standards in Geomatics. ACM-GIS, Washington DC, USA, 36–46, November 10th–11th (2000)
9. Brodeur, J., Bédard, Y., Edwards, G., Moulin, B.: Revisiting the Concept of Geospatial Data Interoperability within the Scope of Human Communication Processes. Transactions in GIS **7**(2), 243–265 (2003)
10. Filho, J.L., Sodre, V.D.F., Daltio, J., Rodrigues Junior, M.F., Vilela, V.: A CASE Tool for Geographic Database Design Supporting Analysis Patterns. Lecture Notes in Computer Science, vol. 3289, pp. 43–54. Springer-Verlag, Berlin/Heidelberg (2004)
11. Fowler, M.: UML 2.0, p. 165. Campus Press, Pearson Education, Paris (2004)
12. Frankel, D.S., Harmon, P., Mukerji, J., Odell, J., Owen, M., Rivitt, P., Rosen, M., Soley, R.: The Zachman Framework and the OMG's Model Driven Architecture. Business Process Trends, Whitepaper (2003)
13. Miller, J., Mukerji, J.: MDA Guide Version 1.0. In: Miller, J., Mukerji, J. (eds.), OMG Document: omg/2003-05-01. http://www.omg.org/mda/mda_files/MDA_Guide_Version1-0.pdf (2003)
14. Miralles, A.: Ingénierie des modèles pour les applications environnementales. Ph.D. Thesis, Université des Sciences et Techniques du Languedoc, p. 338. Département d'informatique, Montpellier, France (2006)
15. Naiburg, E.J., Maksimchuk, R.A.: UML for Database Design. Addison-Wesley, p. 300 (2001)
16. Pantazis, D., Donnay, J.P.: La conception SIG: méthode et formalisme, p. 343. Hermès, Paris (1996)

17. Parent, C., Spaccapietra, S., Zimányi, E.: Conceptual Modeling for Traditional and Spatio-Temporal Applications: The MADS Approach, p. 466. Springer, Berlin/Heidelberg (2006)
18. Sowa, J.F., Zachman, J.A.: Extending and Formalizing the Framework for Information Systems Architecture. IBM Syst. J. **31**(3). IBM Publication G321–5488 (1992)
19. Tryfona, N., Price, R., Jensen, C.S.: Conceptual Models for Spatio-Temporal Applications. In Spatiotemporal Databases: The CHOROCHRONOS approach, Lecture Notes in Computer Science, vol. 2520, Chapter 3. Springer-Verlag, Berlin/Heidelberg (2003)

Modifiable Areal Unit Problem

► Error Propagation in Spatial Prediction

Monitoring

► Data Collection, Reliable Real-Time

Monte Carlo Simulation

► Uncertain Environmental Variables in GIS

Moran Coefficient

► Moran's *I*

Moran's *I*

XIAOBO ZHOU, HENRY LIN
Department of Crop and Soil Sciences, The Pennsylvania State University, University Park, PA, USA

Synonyms

Moran's index; Moran coefficient

Definition

Moran's I, based on cross-products, measures value association and is calculated for n observations on a variable x at locations i, j as

$$I = \frac{\sum_i \sum_{j \neq i} w_{ij}(x_i - \bar{x})(x_j - \bar{x})}{S^2 \sum_i \sum_{j \neq i} w_{ij}} .$$

Where x_i denotes the observed value at location i, $\bar{x}$ is the mean of the x variable over the n locations,

$$S^2 = \frac{1}{n} \sum_i (x_i - \bar{x})^2 ,$$

and w_{ij} is the element of the spatial weights matrix for locations i and j, defined as 1 if location i is contiguous to location j and 0 otherwise. Other more complicated definitions of spatial weights matrices allow for the computation of the Moran's I at various levels of proximity or distance.

Main Text

Moran's I is one of the oldest indicators of spatial autocorrelation and is still a widely accepted measure for determining spatial autocorrelation. It is used to estimate the strength of interdependence between observations of the variable of interest as a function of the distance by comparing the value of x_i at location i with the value x_j at all other locations ($j \neq i$). Moran's I varies from -1 to 1. Positive signage represents positive spatial autocorrelation, while the negative signage represents negative spatial autocorrelation. The Moran's I will approach zero for a large sample size when the variable values are randomly distributed and independent in space.

Cross References

► Autocorrelation, Spatial

M

Moran's Index

► Moran's *I*

Motion Patterns

► Movement Patterns in Spatio-temporal Data

Motion Tracking

► Moving Object Uncertainty

Moved Planning Process

► Participatory Planning and GIS

Movement

► Temporal GIS and Applications

Movement Patterns in Spatio-temporal Data

JOACHIM GUDMUNDSSON[1], PATRICK LAUBE[2], THOMAS WOLLE[1]
[1] NICTA, Sydney, NSW, Australia
[2] Department of Geomatics, University of Melbourne, Melbourne, VIC, Australia

Synonyms

Motion patterns; Trajectory patterns; Exploratory data analysis; Flocking; Converging; Collocation, spatio-temporal; Indexing trajectories; TPR-trees; R-tree, multi-version; Indexing, parametric space; Indexing, native space; Association rules, spatio-temporal; Pattern, moving cluster; Pattern, periodic; Pattern, leadership; Pattern, flock; Pattern, encounter

Definition

Spatio-temporal data is any information relating space and time. This entry specifically considers data involving point objects moving over time. The terms *entity* and *trajectory* will refer to such a point object and the representation of its movement, respectively. *Movement patterns* in such data refer to (salient) events and episodes expressed by a set of entities.

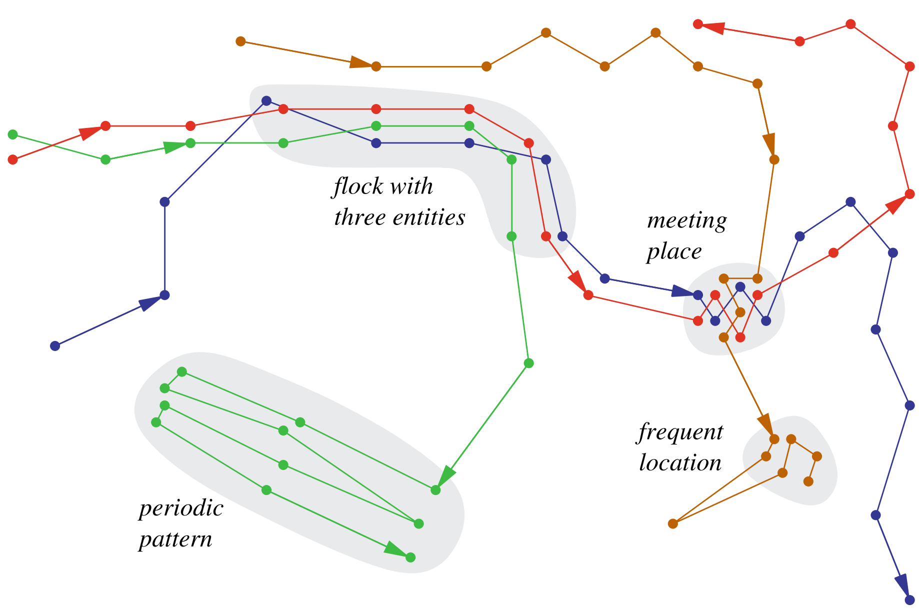
In the case of moving animals, movement patterns can be viewed as the spatio-temporal expression of behaviors, as for example in flocking sheep or birds assembling for the seasonal migration. In a transportation context, a movement pattern could be a traffic jam.

Only formalized patterns are detectable by algorithms. Hence, movement patterns are modeled as any arrangement of subtrajectories that can be sufficiently defined and formalized, see for example the patterns illustrated in Fig. 1. A pattern usually involves a certain number of entities. Furthermore a pattern starts and ends at certain times (temporal footprint), and it might be restricted to a subset of space (spatial footprint).

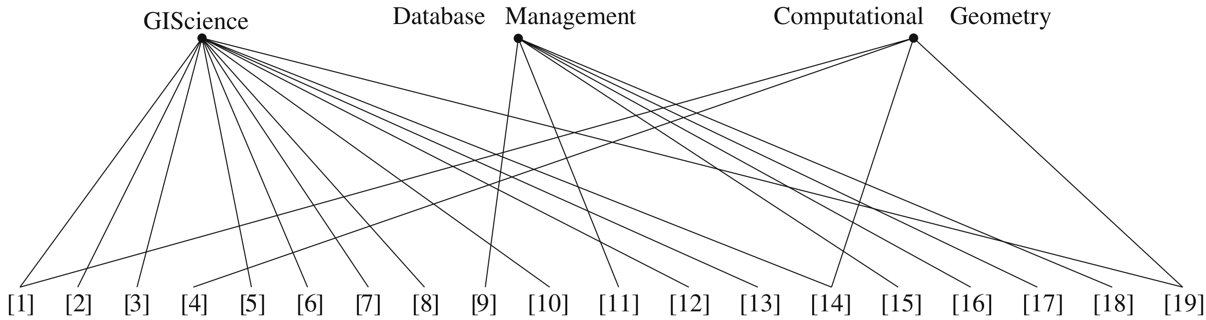
Historical Background

The analysis of movement patterns in Spatio-temporal data is for two main reasons a relatively young and little developed research field. First, emerging from static cartography, geographical information science and theory struggled for a long time with the admittedly substantial challenges of handling dynamics. For many years, occasional changes in a cadastral map were challenging enough, not to mention the constant change of location as is needed for modeling movement.

Second, only in recent years has the technological advancement in tracking technology reached a level that allowed the seamless tracking of individuals needed for the analysis of movement patterns. For many years, the tracking of movement entities has been a very cumbersome and costly undertaking. Hence, movement patterns could only be addressed for single individuals or very small groups. Hägerstrand's time geography [10] may serve as a starting point of a whole branch of geographical information science representing individual trajectories in 3D. The two



Movement Patterns in Spatio-temporal Data, Figure 1 Illustrating the trajectories of four entities moving over 20 time steps. The following patterns are highlighted: a *flock* of three entities over five time-steps, a *periodic pattern* where an entity shows the same Spatio-temporal pattern with some periodicity, a *meeting place* where three entities meet for four time steps, and finally, a *frequently visited location* which is a region where a single entity spends a lot of time



Movement Patterns in Spatio-temporal Data, Figure 2 Access guide to the references in the recommended reading section below

spatial dimensions combined with an orthogonal temporal axis proved to be a very powerful concept for exploring various kinds of spatio-temporal relationships, including movement patterns.

With GPS and various other tracking technologies movement pattern research entered a new era, stepping from 'thread trailing' and 'mark and recapture' approaches to low cost, almost continuous capture of individual trajectories with possibly sub-second sampling rates. Within a few years the situation completely reversed from a notorious data deficit to almost a data overkill, with a lack of suited analytical concepts coping with the sudden surge of movement data. Consequently, the huge potential of analyzing movement in spatio-temporal data has recently attracted the interest of many research fields, both in theory and application, as is outlined in the next two sections.

Scientific Fundamentals

Assume that the entities in Fig. 1 are sheep on a pasture and that they are observed by a geographer, a database expert and a computational geometer. Even though all three experts see the very same sheep, they may all perceive totally different things. The geographer might interpolate a sheep density surface of the pasture. For the database expert in contrast, each sheep may represent a leaf in a dynamic tree optimized for fast queries. Finally, the computational geometer might triangulate the sheep locations in order to detect a flocking pattern. Even though the sheep will not care, their grazing challenges various research fields handling spatio-temporal data. The following overview bundles the different perspectives addressing movement patterns into the three sections exploration, indexing and data mining. See Fig. 2 for a comprehensible access guide to recommended reading.

GIScience: Exploratory Data Analysis and Visualization

In GIScience the term 'pattern' is used in various contexts and meanings when addressing movement. However, as a common denominator, movement patterns are generally conceptualized as salient movement events or episodes in the geospatial representation of moving entities. Given GIScience's legacy in cartography, it is not surprising that movement patterns are often addressed by a combination of geovisualisation and exploration. Exploratory analysis approaches combine the speed and patience of computers with the excellent capability of humans to detect the expected and discover the unexpected given an appropriate graphical representation.

Salient movement patterns may emerge from (i) two-dimensional maps of fixes aligned in trajectories, (ii) movie-like animated maps or even (iii) three-dimensional representations of movement, if time is used as a third, orthogonal axis.

(i) Basic movement patterns are obvious from simple plotting of movement trajectories on a two-dimensional map. Trajectories bundled in narrow, directed bottlenecks represent often used corridors. Less focused trajectory footprints represent more arbitrary movement, such as in grazing animals or visitors at a sports event strolling around a stadium. The application of GUS analysis tools on points and lines representing moving entities has proven to be a very effective approach. For example, GIS tools for generalization, interpolation and surface generation may be applied to support the detection of movement patterns in trajectory data. Brillinger et al. [5] use a regularly sampled vector field to illustrate the overall picture of animals moving in their habitat, with each vector coding in orientation and size for mean azimuth and mean speed at that very location. Dykes and Mountain [8] use a continuous density surface and a 'spotlight' metaphor for the detection of activity patterns. Again, common GIS tools such as algorithms initially designed for the analysis of digital terrain models can easily be adopted for the search for salient movement patterns, for instance to identify 'peaks' of frequent visitation and 'ridges' of busy corridors [8].

(ii) Animation is suited to uncover specific movement behaviors of individuals and groups. Animating moving entities with a constant moving time window in the so-called dynamic view uncovers speed patterns of individuals [2,8]. Flocking or converging are more complex patterns of coordination in groups. Such group patterns are very striking when animating even large numbers or individuals in a movie-like animation.

(iii) The extension of a two-dimensional map with a third orthogonal time axis produces a very powerful tool for uncovering movement patterns. Such ideas go back to Hägerstrand's time geography [10] and have often been adopted in present day geocomputation [12]. In the specific geometry in such a three-dimensional space-time aquarium episodes of immobility and certain speed behaviors produce distinctive patterns of vertical and inclined time lines, respectively. Furthermore, patterns of spatio-temporal collocation can be identified from vertical bottleneck structures in sets of time lines [12].

Indexing Spatio-temporal Trajectories

In the database community considerable research has been focusing on spatial and temporal databases. Research in the spatio-temporal area in many ways started with the dissertations by Lorentzos [15] in 1988 and Langran [13] in 1989. Not surprisingly research has mainly focused on indexing databases so that basic queries concerning the data can be answered efficiently. The most common queries considered in the literature are variants of nearest neighbor queries and range searching queries. For example:

- Spatio-temporal range query, e. g. 'Report all entities that visited region S during the time interval $[t_1, t_2]$.'
- Spatial nearest neighbors given a time interval, e. g. 'Report the entity closest to point p at time t.'
- Temporal nearest neighbors given a spatial region, e. g. 'Report the first entity visiting region S.'

In general one can classify indexing methods used for spatio-temporal data into Parametric Space Indexing methods (PSI) and Native Space Indexing methods (NSI). The PSI method uses the parametric space defined by the movement parameters, and is an efficient approach especially for predictive queries. A typical approach, described by Săltenis et al. [17] is to represent movement defined by its velocity and projected location along each spatial dimension at a global time reference. The parametric space is then indexed by a new index structure referred to as the TPR-tree (Time Parametrized *R*-tree). The TPR-tree is a balanced, multi-way tree with the structure of an *R*-tree. Entries in leaf nodes are pairs of the position of a moving point and a pointer to the moving point, and entries in internal nodes are pairs of a pointer to a subtree and a rectangle that bounds the positions of all moving points or other bounding rectangles in that subtree. The position of a moving point is represented by a reference position and a corresponding velocity vector. To bound a group of d-dimensional moving points, d-dimensional bounding rectangles are used that are also time parametrized, i. e. their coordinates are functions of time. A time-parametrized bounding rectangle bounds all enclosed points or rectangles at all times not earlier than the current time. The search algorithm for a range query also performs computation on the native space by checking the overlap between the range of the query and the trapezoid representation of the node.

The NSI methods represent movement in d dimensions as a sequence of line segments in $d+1$ dimensions, using time as an additional dimension, see for example the work by Hadjieleftheriou et al. [9]. A common approach is to use a multi-dimensional spatial access method like the *R*-tree. An *R*-tree would approximate the whole spatio-temporal evolution of an entity with one Minimum Bounding Region (MBR) that tightly encloses all the locations occupied by the entity during its lifetime. An improvement for indexing movement trajectories is to use a multi version index, like the Multi Version *R*-tree (MVR-tree), also known as a persistent *R*-tree. This index stores all the past states of the data evolution and allows updates to the most recent state. The MVR-tree divides long-lived entities into smaller intervals by introducing a number of entity copies. A query is directed to the exact state acquired by the structure at the time that the query refers to; hence, the cost of answering the query is proportional to the number of entities that the structure contained at that time.

Algorithms and Data Mining

In the previous section different indexing approaches were discussed. This section will focus on mining trajectories for spatio-temporal patterns. This has mainly been done using algorithmic or data mining approaches.

The most popular tools used in the data mining community for spatio-temporal problems has been association rule mining (ARM) and various types of clustering. Association rule mining seeks to discover associations among transactions within relational databases. An association rule is of the form $X \Rightarrow Y$ where X (antecedents) and Y (consequents) are disjoint conjunctions of attribute-value pairs. ARM uses the concept of *confidence* and *support*. The confidence of the rule is the conditional probability of Y given X, and the support of the rule is the prior probability of X and Y.

The probability is usually the observed frequency in the data set. Now the ARM problem can be stated as follows. Given a database of transactions, a minimal confidence threshold and a minimal support threshold, find all association rules whose confidence and support are above the corresponding thresholds.

Chawla and Verhein [18] defined spatio-temporal association rules (STARs) that describe how entities move between regions over time. They assume that space is partitioned into regions, which may be of any size and shape. The aim is to find interesting regions and rules that predict how entities will move through the regions. A region is interesting when a large number of entities leaves (sink), a large number of entities enters (source) or a large number of entities enters and leaves (thoroughfare).

A STAR $(r_i, T_1, q) \Rightarrow (r_j, T_2)$ denotes a rule where entities in a region r_i satisfying condition q during time interval T_1 will appear in region r_j during time interval T_2. The support of a rule δ is the number, or ratio, of entities that follow the rule. The *spatial* support takes the size of the involved regions into consideration. That is, a rule with support s involving a small region will have a larger spatial support than a rule with support s involving a larger region. Finally, the confidence of a rule δ is the conditional probability that the consequent is true given that the antecedent is true. By traversing all the trajectories all possible movements between regions can be modeled as a rule, with a spatial support and confidence. The rules are then combined into longer time intervals and more complicated movement patterns.

Some of the most interesting spatio-temporal patterns are periodic patterns, e. g. yearly migration patterns or daily commuting patterns. Mamoulis et al. [16] considered the special case when the period is given in advance. They partition space into a set of regions which allows them to define a pattern P as a τ-length sequence of the form r_0, $r_1, \ldots, r_{\tau-1}$, where r_i is a spatial region or the special character *, indicating the whole spatial universe. If the entity follows the pattern enough times, the pattern is said to be *frequent*. However, this definition imposes no control over the density of the regions, i. e. if the regions are too large then the pattern may always be frequent. Therefore an additional constraint is added, namely that the points of each subtrajectory should form a cluster inside the spatial region.

Kalnis et al. [11] define and compute moving clusters where entities might leave and join during the existence of a moving cluster. For each fixed discrete time-step t_i they use standard clustering algorithms to find clusters with a minimum number of entities and a minimum density. Then they compare any cluster c found for t_i with any (moving) cluster c' found for time-step t_{i-1}. If c and c' have enough entities in common, which is formally specified by a threshold value, then c' can be extended by c, which results in a *moving cluster*. They propose several ideas to increase the speed of their method, e. g. by avoiding redundant cluster comparisons, or approximating moving clusters instead of giving exact solutions, and they experimentally analyze their performance.

In 2004 Laube et al. [14] defined a collection of spatio-temporal patterns based on direction of movement and location, e. g. flock, leadership, convergence and encounter, and they gave algorithms to compute them efficiently. As a result there were several subsequent articles studying the discovery of these patterns. Benkert et al. [4] modified the original definition of a flock to be a set of entities moving close together during a time interval. Note that in this definition the entities involved in the flock must be the same during the whole time interval, in contrast to the moving cluster definition by Kalnis et al. [11]. Benkert et al. [4] observed that a flock of m entities moving together during k time steps corresponds to a cluster of size m in $2k$ dimensional space. Thus the problem can be restated as clustering in high dimensional space. To handle high dimensional space one can use well-known dimensionality reduction techniques. There are several decision versions of the problem that have been shown to be NP-hard, for example deciding if there exists a flock of a certain size, or of a certain duration. The special case when the flock is stationary is often called a *meeting* pattern.

Andersson et al. [1] gave a more generic definition of the pattern *leadership* and discussed how such leadership patterns can be computed from a group of moving entities. The proposed definition is based on behavioral patterns discussed in the behavioral ecology literature. The idea is to define a leader as an entity that (1) does not follow anyone else, (2) is followed by a set of entities and (3) this behavior should continue for a duration of time. Given these rules all leadership patterns can be efficiently computed.

Be it exploratory analysis approaches, indexing techniques or data mining algorithms, all effort put in theory ultimately leads to more advanced ways of inferring high level process knowledge from low level tracking data. The following section will illustrate a wide range of fields where such fundamentals underlie various powerful applications.

Key Applications

Animal Behavior

The observation of behavioral patterns is crucial to animal behavior science. So far, individual and group patterns are rather directly observed than derived from tracking data. However, there are more and more projects that collect animal movement by equipping them with GPS-GSM collars. For instance, since 2003 the positions of 25 elks in Sweden are obtained every 30 minutes. Other researchers attached small GPS loggers to racing pigeons and tracked their positions every second during a pigeon's journey. It is even possible to track the positions of insects, e. g. butterflies or bees, however most of the times non-GPS based technologies are used that allow for very small and light sensors or transponders. Analyzing movement patterns of animals can help to understand their behavior in many different aspects. Scientists can learn about places that are popular for individual animals, or spots that are frequented by many animals. It is possible to investigate social interactions, ultimately revealing the social structure within a group of animals. A major focus lies on the investigation of leading and following behavior in socially interacting animals, such as in a flock of sheep or a pack of wolves [7]. On a larger scale, animal movement data reflects very well the seasonal or permanent migration behavior. In the animation industry, software agents implement movement patterns in order to realistically mimic the behavior of animal groups. Most prominent is the flocking model implemented in NetLogo which mimics the flocking of birds [19].

Human Movement

Movement data of people can be collected and used in several ways. For instance, using mobile phones that communicate with a base station is one way to gather data about the approximate locations of people. Traffic-monitoring devices such as cameras can deliver data on the movement of vehicles. With the technological advancement of mobile and position aware devices, one could expect that tracking data will be increasingly collectable. Although tracking data of people might be available in principle, ethical and privacy aspects need to be taken into consideration before gathering and using this data [6]. Nonetheless, if the data is available, it could be used for urban planning, e. g. to plan where to build new roads or where to extend public transport.

The detection of movement patterns can furthermore be used to optimize the design of location-based-services (LBS). The services offered to a moving user could not only be dependent on the actual position, but also on the estimated current activity, which may be derived from a detected movement pattern.

Traffic Management

Movement patterns are used for traffic management in order to detect undesirable or even dangerous constellations of moving entities, such as traffic jams or airplane course conflicts. Traffic management applications may require basic Moving Object Database queries, but also more sophisticated movement patterns involving not just location but also speed, movement direction and other activity parameters.

Surveillance and Security

Surveillance and intelligence services might have access to more detailed data sets capturing the movement of people, e. g. coordinates from mobile phones or credit card usage, video surveillance camera footage or maybe even GPS data. Apart from analyzing the movement data of a suspect to help prevent further crime, it is an important task to analyze the entire data set to identify suspicious behavior in the first place. This leads to define 'normal behavior' and then search the data for any outliers, i. e. entities that do not show normal behavior. Some specific activities and the corresponding movement patterns of the involved moving entities express predefined signatures that can be automatically detected in spatio-temporal or footage data. One example is that fishing boats in the sea around Australia have to report their location in fixed intervals. This is important for the coast guards in case of an emergency, but the data can also be used to identify illegal fishing in certain areas. Another example is that a car thief is expected to move in a very characteristic and hence detectable way across a surveilled car park. Movement patterns have furthermore attracted huge interests in the field of spatial intelligence and disaster management. Batty et al. [3] investigated local pedestrian movement in the context of disaster evacuation where movement patterns such as congestion or crowding are key safety issues.

Military and Battlefield

The digital battlefield is an important application of moving object databases. Whereas real-time location data of friendly troops is easily accessible, the enemy's location may be obtained from reconnaissance planes with only little time lag. Moving object databases not only allow the dynamic updating of location and status of tanks, airplanes and soldiers, but also answering spatio-temporal queries and detecting complex movement patterns. Digital battlefield applications answer spatio-temporal range queries like 'Report all friendly tanks that are currently in region *S*.' A more complex movement pattern in a digital battlefield context would be the identification of the convergence area where the enemy is currently concentrating his troops.

Sports Scene Analysis

Advancements in many different areas in technology are also influencing professional sports. For example, some of the major tennis tournaments provide three-dimensional reconstructions of every single point played, tracking the players and the balls. It is furthermore known that, e. g. football coaches routinely analyze match video archives to learn about an opponents behaviors and strategies. Making use of tracking technology, the movement of the players and the ball can be described by 23 trajectories over the length of the match. Researchers were able to develop a model that is based on the interactions between the players and the ball. This model can be used to quantitatively express the performance of players, and more general, it might lead to an improved overall strategy. Finally, real-time tracking systems are developed that keep track of both players and the ball in order to assist the referee with the detection of the well-defined but nevertheless hard to perceive offside pattern.

Movement in Abstract Spaces

In contrast to tracking and analyzing the movement of animals and people on the surface of the earth, it is also possible to obtain and analyze spatio-temporal data in abstract spaces also in higher dimensions. Every scatter plot that constantly updates the changes in the x and y values, produces individual trajectories open for movement analysis. Two stock exchange series plotted against each other could build such a dynamic scatter-plot. As another example, basic ideological conflicts can be used to construct abstract ideological spaces. Performing factor analysis on referendum data, researchers hypothesized a structure of mentality consisting of dimensions such as 'political left vs. political right' or 'liberal vs. conservative'. Whole districts or even individuals such as members of parliament could now be localized and re-localized in such ideological space depending on their voting behavior and its change over time, respectively. Movement in such a space represents the change of opinions and analyzing this can lead to more insight and understanding of human psychology and politics.

Future Directions

For simplicity reasons, theory and application of movement patterns in spatio-temporal data focused so far largely on moving point objects. However, many processes can only be modeled as dynamics in fields or in their discredited counterparts that is dynamic polygons. When monitoring a hurricane threatening urban areas, the tracking of its eye alone may not provide sufficient information, but additional tracking of its changing perimeter will be required. The consideration of both location and change of polygonal objects raises the conceptualization and detection of movement patterns to a higher level of complexity, which has only rarely been addressed so far.

For the many fields interested in movement, the overall challenge lies in relating movement patterns with the underlying geography, in order to understand where, when and ultimately why the entities move the way they do. Grazing sheep, for example, may perform a certain movement pattern only when they are on a certain vegetation type. Homing pigeons may show certain flight patterns only when close to a salient landscape feature such as a river or a highway. And, the movement patterns expressed by a tracked vehicle will obviously be very dependent on the environment the vehicle is moving in, be it in a car park, in a suburb or on a highway. Thus, patterns have to be conceptualized that allow linking of the movement with the embedding environment.

Cross References

► Indexing, Query and Velocity-Constrained
► Privacy Threats in Location-Based Services

Recommended Reading

1. Andersson, M., Gudmundsson, J., Laube, P., Wolle T.: Reporting leadership patterns among trajectories. In: Proceedings of the 2007 ACM Symposium on Applied Computing, pp. 3–7. ACM Press, New York (2007)
2. Andrienko, N.V., Andrienko, G.L.: Interactive maps for visual data exploration. Int. J. Geogr. Inf. Sci. **13**(4), 355–374 (2003)
3. Batty, M.,Desyllas, J., Duxbury, E.: The discrete dynamics of small-scale spatial events: agent-based models of mobility in carnivals and street parades. Int. J. Geogr. Inf. Sci. **17**(7), 673–697 (2003)
4. Benkert, M., Gudmundsson, J., Hübner, F., Wolle, T.: Reporting flock patterns. In: Proceedings of the 14th European Symposium on Algorithms. Lecture Notes in Computer Science, vol. 4168 pp. 660–671. Springer, Berlin, Heidelberg (2006)
5. Brillinger, D.R., Preisler, H.K., Ager, A.A., Kie, J.G.: An exploratory data analysis (EDA) of the paths of moving animals. J. Stat. Plan. Inf. **122**(2), 43–63 (2004)
6. Dobson, J.E., Fisher, P.F.: Geoslavery. IEEE Technology and Society Magazine **22**(1), 47–52 (2003)
7. Dumont, B., Boissy, A., Achard, C., Sibbald, A.M., Erhard, H.W.: Consistency of animal order in spontaneous group movements allows the measurement of leadership in a group of grazing heifers. Appl. Anim. Behav. Sci. **95**(1–2), 55–66 (2005)
8. Dykes, J.A., Mountain, D.M.: Seeking structure in records of spatio-temporal behaviour: visualization issues, efforts and application. Comput. Stat. Data Anal. **43**(4), 581–603 (2003)
9. Hadjieleftheriou, M., Kollios, G., Tsotras, V.J., Gunopulos, D.: Indexing spatio-temporal archives. The VLDB J. **15**(2), 143–164 (2006)
10. Hägerstrand, T.: What about people in regional science. Pap. Region. Sci. Assoc. **24**, 7–21 (1970)
11. Kalnis, P., Mamoulis, N., Bakiras, S.: On discovering moving clusters in spatio-temporal data. In: Medeiros, C.B., Egenhofer, M.J., Bertino, E. (eds.) Proceedings of the 9th International Symposium on Advances in Spatial and Temporal Databases. Lecture Notes in Computer Science, vol. 3633, pp. 364–381. Springer, Berlin, Heidelberg (2005)
12. Kwan, M.P.: Interactive geovisualization of activity-travel patterns using three dimensional geographical information systems:

M

a methodological exploration with a large data set. Transportation Research Part C **8**(1–6), 185–203 (2000)
13. Langran, G.: Time in Geographic Information Systems. PhD thesis, University of Washington (1999)
14. Laube, P., van Kreveld, M., Imfeld, S.: Finding REMO – detecting relative motion patterns in geospatial lifelines. In: Fisher, P.F. (ed.) Developments in Spatial Data Handling, Proceedings of the 11th International Symposium on Spatial Data Handling, pp. 201–214. Springer, Berlin Heidelberg (2004)
15. Lorentzos, N.A.: A Formal Extension of the Relational Model for the Representation and Manipulation of Generic Intervals. PhD thesis, Birbeck College, University of London (1988)
16. Mamoulis, N., Cao, H., Kollios, G., Hadjieleftheriou, M., Tao, Y., Cheung, D.: Mining, indexing, and querying historical spatiotemporal data. In: Proceedings of the 10th ACM International Conference On Knowledge Discovery and Data Mining, pp. 236–245. ACM Press, New York (2004)
17. Săltenis, S., Jensen, C.S., Leutenegger, S.T., Lopez, M.A.: Indexing the positions of continuously moving objects. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 331–342 (2000)
18. Verhein, F., Chawla, S.: Mining spatio-temporal association rules, sources, sinks, stationary regions and thoroughfares in object mobility databases. In: Proceedings of the 11th International Conference on Database Systems for Advanced Applications. Lecture Notes in Computer Science, vol. 3882, pp. 187–201. Springer, Berlin, Heidelberg (2006)
19. Wilensky, U.: Netlogo flocking model. http://ccl.northwestern.edu/netlogo/models/Flocking

Moving Average (MA) Process Model

► Spatial Regression Models

Moving Average Regression

► Spatial and Geographically Weighted Regression

Moving Object Constraint Databases

► Constraint Databases and Moving Objects

Moving Object Databases

► Moving Object Uncertainty

Moving Object Languages

RALF HARTMUT GÜTING
Faculty for Mathematics and Computer Science,
Distance University of Hagen, Hagen, Germany

Synonyms

Query languages for moving objects

Definition

The term refers to query languages for *moving objects databases*. Corresponding database systems provide concepts in their data model and data structures in the implementation to represent moving objects, i. e., continuously changing geometries. Two important abstractions are *moving point*, representing an entity for which only the time-dependent position is of interest, and *moving region*, representing an entity for which also the time-dependent shape and extent are relevant. Examples of moving points are cars, trucks, airplanes, ships, mobile phone users, RFID-equipped goods, and polar bears; examples of moving regions are forest fires, deforestation of the Amazon rain forest, oil spills in the sea, armies, epidemic diseases, and hurricanes.

There are two flavors of such databases. The first, which is called the *location management* perspective, represents information about a set of currently moving objects. Basically, one is interested in efficiently maintaining their locations and asking queries about the current and expected near future positions and relationships between objects. In this case, no information about histories of movement is kept. The second is called the *spatio-temporal data* perspective; here the complete histories of movements are represented. The goal in the design of query languages for moving objects is to be able to ask any kind of question about such movements, perform analyses, and derive information in a way that is as simple and elegant as possible. Such queries must be executed efficiently.

Historical Background

The field of moving objects databases, with the related query languages, came into being in the late 1990s mainly by two parallel developments. First, the Wolfson group developed a model in a series of papers [13,15,16,17] that allows one to keep track in a database of a set of time-dependent locations, e. g., to represent vehicles. They observed that one should store in a database not the locations directly, which would require high update rates, but rather a motion vector, representing an object's expected position over time. An update to the database is needed only when the deviation between the expected position and the real position exceeds some threshold. At the same time this concept introduces an inherent, but bounded uncertainty about an object's real location. The group formalized this model introducing the concept of a *dynamic attribute*. This is an attribute of a normal data type which changes implicitly over time. This implies that results of queries over such attributes also change implicitly over time. They introduced a related query language called future temporal logic (FTL) that allows one to specify time-dependent relationships between expected positions of moving objects. Hence this group established the location-management perspective.

Second, the European project CHOROCHRONOS set out to integrate concepts from spatial and temporal databases and explored the *spatio-temporal data* perspective. This means, one represents in a database time-dependent geometries of various kinds such as points, lines, or regions. Earlier work on spatio-temporal databases had generally admitted only discrete changes. This restriction was dropped and continuously changing geometries were considered. Güting and colleagues developed a model based on the idea of *spatio-temporal data types* to represent histories of continuously changing geometries [5,9,6,2]. The model offers data types such as *moving point* or *moving region* together with a comprehensive set of operations. For example, there are operations to compute the projection of a moving point into the plane, yielding a *line* value, or to compute the distance between a moving point and a moving region, returning a time dependent real number, or *moving real*, for short. Such data types can be embedded into a DBMS data model as attribute types and can be implemented as an extension package.

A second approach to data modeling was pursued in CHOROCHRONOS by Grumbach and colleagues who applied the constraint model to the representation of moving objects [7,11] and implemented a prototype called Dedale. Constraint databases can represent geometries in *n*-dimensional spaces; since moving objects exist in 3D (2D + time) or 4D (3D + time) spaces, they can be handled by this approach. Several researchers outside CHOROCHRONOS also contributed to the development of constraint-based models for moving objects.

Scientific Fundamentals

The following two subsections describe two major representations for the location management and the spatio-temporal data flavor of moving objects databases in some detail, namely the MOST model and FTL language, and the approach of spatio-temporal data types. In a short closing subsection we mention some further work related to languages for moving objects.

Modeling and Querying Current Movement – the MOST Model and FTL Language

In this section we discuss moving objects databases based on the location management perspective and a related query language. That is, the database keeps track of a collection of objects moving around currently and we wish to be able to answer queries about the current and expected near-future positions. Such sets of entities might be taxi cabs in a city, trucks of a logistics company, or military vehicles in a military application. Possible queries might be:

- Retrieve the three free cabs closest to Cottle Road 52 (a passenger request position).
- Which trucks are within 10 km of truck T70 (which needs assistance)?
- Retrieve the friendly helicopters that will arrive in the valley within the next 15 min and then stay in the valley for at least 10 min.

Statically, the positions of a fleet of taxi cabs, for example, could be easily represented in a relation

```
taxi cabs(id: int, pos: point).
```

Unfortunately this representation needs frequent updates to keep the deviation between real position and position in the database small. This is not feasible for large sets of moving objects.

The moving objects spatio-temporal (MOST) data model [13,16] discussed in this section stores, instead of absolute positions, a motion vector which represents a position as a linear function of time. This defines an expected position for a moving object. The distance between the expected position and the real position is called the *deviation*. Furthermore, a *distance threshold* is introduced and a kind of contract between a moving object and the database server managing its position is assumed. The contract requires that the moving object observes the deviation and sends an update to the server when it exceeds the threshold. Hence the threshold establishes a bound on the *uncertainty* about an object's real position.

The MOST model relies on a few basic assumptions: A database is a set of object classes. Each object class is given by its set of attributes. Some spatial data types like *point*, *line*, or *polygon* with suitable operations are available. Object classes may be designated as spatial which means they have a single spatial attribute. Spatial operations can then be directly applied to objects, e. g., *distance* (o_1, o_2) for two objects o_1 and o_2. Besides object classes, the database contains an object called *Time* which yields the current time at every instant. Time is assumed to be discrete and can be represented by *integer* values. The value of the *Time* object increases by one at each clock tick (e. g., every second).

M

Dynamic Attributes

A fundamental new concept in the MOST model is that of a *dynamic attribute*. Each attribute of an object class is classified to be either static or dynamic. A dynamic attribute is of a standard data type (e. g., *int*, *real*) within the DBMS conceptual model, but changes its value automatically over time. This means that queries involving such attributes also have time-dependent results, even if time is not mentioned in the query and no updates to the database occur.

For a data type to be eligible for use in a dynamic attribute, it is necessary that the type has a value 0 and an addition operation. This holds for numeric types but can be extended to types like *point*. A dynamic attribute A of type T is then represented by three subattributes $A.value$, $A.updatetime$, and $A.function$, where $A.value$ is of type T, $A.updatetime$ is a time value, and $A.function$ is a function f: $int \to T$ such that at time $t = 0$, $f(t) = 0$. The semantics of this representation is called the value of A at time t and defined as

$$value(A, t) = A.value + A.function(t - A.updatetime). \quad \text{for} \quad t \geq A.updatetime$$

When attribute A is mentioned in a query, its dynamic value $value(A, t)$ is meant.

Representing Object Positions

A simple way of modeling objects moving freely in the *xy*-plane would be to introduce an attribute *pos* with two dynamic subattributes *pos.x* and *pos.y*. For example, we might define an object class for cars:

```
cars(license_plate: string, pos:
(x: dynamic real, y: dynamic real)).
```

For vehicles, a more realistic assumption is that they move along road networks. A more sophisticated modeling of time dependent positions in MOST uses a *loc* attribute with six subattributes *loc.route*, *loc.startlocation*, *loc.starttime*, *loc.direction*, *loc.velocity*, and *loc.uncertainty*. Here *loc.route* is a (pointer to) a *line* value (a polyline) describing the geometry of the road on which the vehicle is moving. Say that on *loc.route*, one chooses a point as origin and a particular direction as positive direction. Then, the initial location *loc.startlocation* is given by its distance from the origin; this distance is positive if the direction from the origin to the initial location is in the positive direction, otherwise it is negative. The velocity also is negative or positive accordingly. *Startlocation*, *starttime*, and *velocity* correspond to the components of a dynamic attribute explained above. The value of *loc* at time t, $value(loc, t)$ is now a position on the *route* polyline defined in the obvious way. Query evaluation may take the *uncertainty* into account.

Semantics of Queries, Query Types

In traditional databases, the semantics of a query are defined with respect to the current state of a database. This is not sufficient for the MOST model, as queries may refer to future states of a database. A *database state* is a mapping that associates each object class in the database with a set of objects of appropriate types, and the *Time* object with a time value. Let $o.A$ and $o.A.B$ denote attribute A of object o and subattribute B of attribute A of object o, respectively. In database state s, the value of $o.A$ is denoted $s(o.A)$ and the value of the *Time* object as $s(Time)$. For each dynamic attribute A, its value in state s is $value(A, s(Time))$.

The semantics of queries are now defined relative to a database history. A *database history* is an infinite sequence of database states, one for each clock tick, beginning at some time u, hence is $s_u, s_{u+1}, s_{u+2}, \ldots$ An update at some time $t > u$ will affect all database states from t on. Hence with each clock tick there is a new database state, and with each update a new database history. Let $Q(H,t)$ denote a query Q evaluated on database history H assuming a current time t.

There are now two types of queries, namely *instantaneous* and *continuous* query.[1] An instantaneous query issued at time t is evaluated once on the history starting at time t, hence:

$$Q(H_t, t). \qquad (\textit{instantaneous query})$$

In contrast, a continuous query is (conceptually) reevaluated once for each clock tick, hence as a sequence of instantaneous queries:

$$Q(H_t, t), Q(H_{t+1}, t + 1), Q(H_{t+2}, t + 2), \ldots \qquad (\textit{continuous query})$$

The result of a continuous query changes over time; at time u the result of $Q(H_u, u)$ is valid. Of course, reevaluating the query on each clock tick is not feasible, instead, the evaluation algorithm for such queries is executed only once and produces a time dependent result in the form of a set of tuples with associated time stamps. Reevaluation is necessary only for explicit updates.

[1] There exists a further query type, persistent query, which is omitted here.

The Language FTL

The query language associated with the MOST model is called FTL. The following are example queries formulated in FTL.

1. Which trucks are within 10 km of truck T70?
```
RETRIEVE t
FROM trucks t, trucks s
WHERE s.id = 'T70' ∧ dist(s, t) <= 10
```
Here nothing special happens, yet, the result is time dependent.
2. Retrieve the helicopters that will arrive in the valley within the next 15 min and then stay in the valley for at least 10 min.
```
RETRIEVE h
FROM helicopters h
WHERE eventually_within_15 (inside(h,
Valley) ∧ always_for_10 (inside(h, Valley))
```
Here *Valley* is a polygon object.

The general form of a query in FTL is[2]

```
RETRIEVE <target-list> FROM <object classes>
  WHERE <FTL-formula>.
```

The interesting part is the FTL formula. FTL formulas are similar to first-order logic, hence they are built from constants, function symbols, predicate symbols, variables and so forth. Some special constructs in the definition of formulas are the following:

- If f and g are formulas, then f **until** g and **nexttime** f are formulas

The semantics of a formula are defined with respect to:

- A variable assignment μ which associates with each variable in the formula a corresponding database object (e. g., for s, t in the first example query $\mu = [(s, T_{10}), (t, T_{20})]$ where T_i are truck objects in the database)
- A database state s on history h

Next, it is necessary to define what it means for a formula to be satisfied at state s on history H with respect to variable assignment μ (satisfied at (s, μ) for short). The semantics of the special constructs are defined as follows:

- f **until** g is satisfied at $(s, \mu) :\Leftrightarrow$ either g is satisfied at (s, μ), or there exists a future state s' on history H such that (g is satisfied at $(s', \mu) \wedge$ for all states s_i on history H before state s', f is satisfied at (s_i, μ)).
- **nexttime** f is satisfied at $(s, \mu) :\Leftrightarrow f$ is satisfied at (s', μ) where s' is the state immediately following s in history H.

Based on these temporal operators with well-defined semantics, some derived notations can be defined:

- **eventually** $g \equiv$ *true* **until** g
- **always** $g \equiv$ (¬**eventually** $(\neg g)$)

[2]In the original literature about FTL, a single class of moving objects is assumed and the FROM clause omitted.

In addition, it is useful to have bounded temporal operators:

- f **until_within_c** g asserts that there exists a future time within c units of time from now such that g holds and until that time f will hold continuously
- f **until_after_c** g asserts that there exists a future time after at least c units of time from now such that g holds and until that time f will hold continuously

Based on these, one can again define further bounded temporal operators:

- **eventually_within_c** $g \equiv$ *true* **until_within_c** g
- **eventually_after_c** $g \equiv$ *true* **until_after_c** g
- **always_for_c** $g \equiv g$ **until_after_c** *true*

That concludes the explanation of the semantics of the constructs used in example query 2 above.

Evaluation

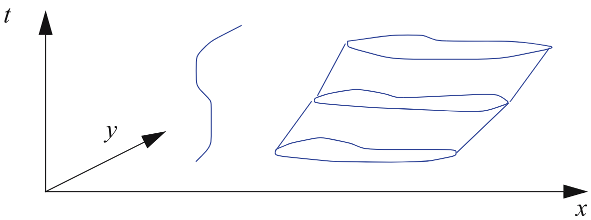
The algorithm for evaluating FTL queries can only be briefly sketched here. The basic idea is to compute a relation for every subformula of the given FTL formula bottom up, starting from atomic formulas like `dist(s, t) <= 10`. The relation R_f for subformula f has an attribute for each free variable occurring in f, and two attributes for time stamps t_{start}, t_{end}. Hence the relation for formula `dist(s, t) <= 10` has schema $(s, t, t_{start}, t_{end})$. It has a tuple (o, o', T, T') for every pair of objects O, O' and for every maximal time interval $[T, T']$ such that objects O, O' are within distance 10 throughout the interval $[T, T']$. For operators combining two subformulas such as $f \wedge g$ or f **until** g it is then possible to compute their relations essentially by joins over common object identifier attributes (equal variables in both subformulas), manipulating the time stamps in an appropriate way.

The result relation for the complete formula is the result for a continuous query. As time progresses, tuples of this relation can be added to or removed from the current result. It can also be used to answer an instantaneous query selecting just the tuples valid at the time of issuing this query.

Modeling and Querying History of Movement – Spatio-temporal Data Types

This section discusses the spatio-temporal data perspective for moving objects databases. That is, we consider the geometries stored in spatial databases and allow them to change continuously over time. Some of the most important abstractions used in spatial databases are

- Point – an object for which only the position in space is relevant
- Line – a curve often representing connections, such as roads and rivers



Moving Object Languages, Figure 1 A moving point and a moving region

- Region – representing objects for which the extent is relevant
- Partition – subdivisions of the plane, e. g. of a country into states
- Network – graph or network structures over roads, rivers, power lines, etc.

Such abstractions are usually captured in *spatial data types*, consisting of the type together with operations.

Spatio-temporal Data Types The idea of the approach [5] presented in the following is to introduce spatio-temporal data types that encapsulate time dependent geometries with suitable operations. For moving objects, point and region appear to be most relevant, leading to data types *moving point* and *moving region*, respectively. The *moving point* type can represent entities such as vehicles, people, or animals moving around whereas the *moving region* type can represent hurricanes, forest fires, armies, or flocks of animals, for example. Geometrically, values of spatio-temporal data types are embedded into a 3D space (2D + time) if objects move in the 2D plane, or in a 4D space if movement in the 3D space is modeled. Hence, a moving point and a moving region can be visualized as shown in Fig. 1.

In the following, we first motivate the approach by introducing an example database with spatio-temporal data types, providing a number of operations on these data types, and forulating queries using these operations. We then discuss the underlying design principles for types and operations, the distinction between abstract model and discrete model in defining the semantics of types, and the structure of type system and operations in more detail. Finally, the implementation strategy is briefly outlined.

Example Operations and Queries As a simple example, suppose we have two relations representing cars and weather conditions, whose movements have been recorded.

```
cars (license_plate: string, trip: mpoint)
weather (id: string, area: mregion)
```

Here *mpoint* and *mregion* are abbreviations for the *moving point* and *moving region* data types, respectively. Assume further, some available operations on these types, used in queries below, are the following:

Signature		Operation
moving(*point*)	→ *line*	**trajectory**
moving(*region*)	→ *region*	**traversed**
moving(α)	→ *periods*	**deftime**
moving(*point*) × *moving*(*region*)	→ *moving*(*point*)	**intersection**
moving(α) × *instant*	→ *intime*(α)	**atinstant**
intime(α)	→ *instant*	**inst**
intime(α)	→ α	**val**
periods	→ *int*	**duration**
int × *int* × *int* × *int*	→ *instant*	**theinstant**

In these signatures, *moving* is viewed as a type constructor that transforms a type α into a time dependent version of that type, *moving*(α). These operations use further data types:

- *instant*, representing an instant of time
- *periods*, representing a set of disjoint time intervals
- *intime*(α), where *intime* is a type constructor building pairs of an *instant* and a value of another type α.

The operations have the following meaning: **Trajectory** and **traversed** compute the projection of a moving point or moving region into the 2D plane; **deftime** computes the projection on the time axis. The intersection of a moving point and a moving region is a moving point again containing the parts of the moving point inside the moving region. **Atinstant** evaluates the moving object at a particular instant of time, returning an (*instant*, α) pair. **Inst** and **val** allow access to the components of *intime*(α) pairs. **Duration** returns the total length of time intervals in a periods value (say, in seconds). Operator **theinstant** constructs instants of time for a variable number of integer arguments (here four) in the order year, month, day, hour, minute, and second returning the first instant of time of such a time interval.

We can then formulate the following queries:

1. What was the route taken by the car "DO-GL 871"?
```
SELECT trajectory(trip) AS route
FROM cars WHERE license_plate = "DO-GL 871"
```
2. What was the total area swept by the fog region with identifier "F276"?
```
SELECT traversed(area) AS fogarea
FROM weather WHERE id = "F276"
```
3. How many cars stayed in the fog area for more than 30 min?
```
SELECT count(*)
FROM cars AS c, weather AS w
WHERE duration(deftime(intersection(c.trip,
   w.area))) > 1800
```

4. Where was the fog area at 5 p.m. (on the respective day January 8, 2007)?

```
SELECT val(atinstant(area,
   theinstant(2007, 1, 8, 17)))
FROM weather WHERE id = "F276"
```

Goals in the Design of Types and Operations The examples have illustrated the basic approach of using spatio-temporal data types. Starting from this idea, the question is how exactly data types and operations should be chosen. Such a systematic design was given in [9] and the types and operations above are already part of that design. The design pursues the following goals:

- *Closure of type system*. Type constructors should be applied systematically and consistently. This means in particular:
  - For all base types of interest, we have related temporal ("moving") types.
  - For all temporal types (whose values are functions from time into some domain), there exist types to represent their projection into domain and range.
- *Genericity*. There will be a large set of data types. Operations should be designed in a generic way to cover as many types as possible.
- *Consistency between nontemporal and temporal types*. The structure of a temporal type, taken at a particular instant of time, should agree with the structure of the corresponding static type.
- *Consistency between nontemporal and temporal operations*. For example,

$$\mathbf{val}(\mathbf{atinstant}(\mathbf{intersection}(mp, mr), t)) =$$
$$\mathbf{intersection}(\mathbf{val}(\mathbf{atinstant}(mp, t)), \mathbf{val}(\mathbf{atinstant}(mr, t))$$

Abstract and Discrete Model Before considering the type system in more detail, one should understand the meaning of data types. There exists a choice at what level we define the semantics of types. For example, a moving point could be viewed in two ways:

- A continuous function from time (viewed as isomorphic to the real numbers) into the Euclidean plane, i. e., a function $f : \mathbb{R} \to \mathbb{R}^2$.
- A polyline in the 3D space representing such a function.

The essential difference is that in the first case, we define the semantics of the type in terms of infinite sets without fixing a finite representation. In the second case we choose a finite representation. We call the first an *abstract model* and the second a *discrete model*. Note that there are many discrete models for a given abstract model. For example, a moving point might also be represented as a sequence of splines. The properties of such models can be summarized as follows:

- Abstract models are mathematically simple, elegant, and uniform, but not directly implementable.
- Discrete models are more complex and heterogeneous, but can be implemented.

As a consequence, the design of spatio-temporal data types proceeds in two steps: First, an abstract model of types and operations is designed. Second, a discrete model to represent (a large part of) the abstract model is constructed.

Type System The structure of the type system is illustrated in Fig. 2. It reflects the design goals stated above. Projections of standard types are represented as sets of disjoint intervals over the respective base type; the *range* type constructor yields such types *range*(α) for base type α. Projections of time dependent geometries can generally be of different data types. For example, a moving point can "jump around", i. e., change position in discrete steps, yielding a *points* value (set of points) as a projection. Or it can move continuously, yielding a *line* value (a curve in the plane). Note that *line* and *region* values can have multiple components, so that the respective projection operations are closed. The *intime*(α) types have been omitted in this figure.

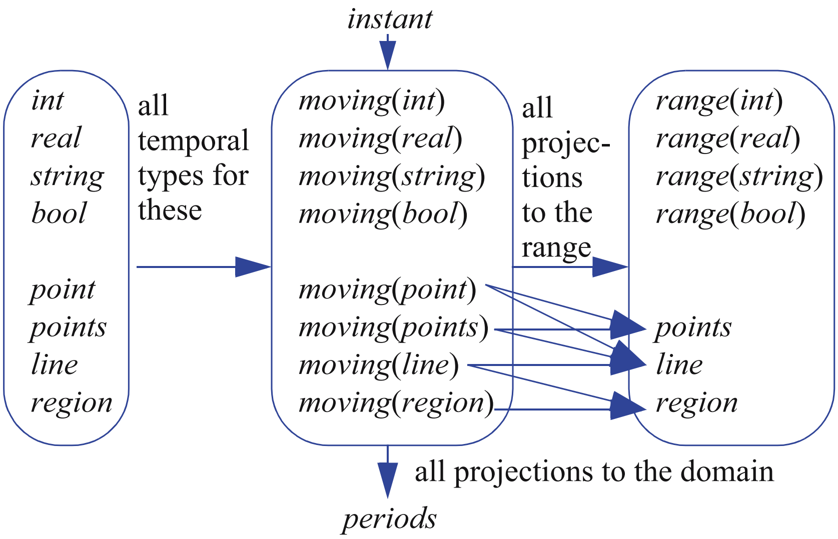
Operations The design of operations proceeds in three steps:

1. Carefully design operations for nontemporal types, using generic definition techniques.
2. By a technique called lifting make them all time dependent in a way consistent with the static definition.
3. Add specialized operations for the temporal types.

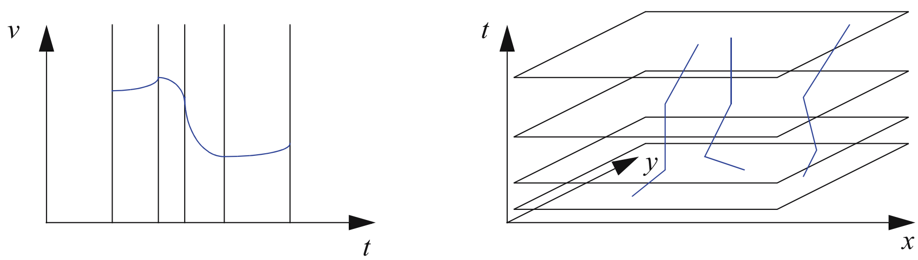
There is a comprehensive set of operations first on the nontemporal types having classes of operations such as predicates, set operations, aggregate, numeric, distance and direction operations. Second, these are all lifted, which means each of their arguments may become time dependent which makes the result time dependent as well. Third, specialized operations on temporal types have classes of operations addressing projection to domain and range, interaction with values in domain and range, and operations to deal with rate of change (e. g., derivative).

Implementation Implementation is based on the discrete model proposed in [6] using algorithms for the operations studied in [2]. The discrete model uses the so-called *sliced representation* as illustrated in Fig. 3.

A temporal function value is represented as a time-ordered sequence of *units* where each unit has an associated time interval and time intervals of different units are disjoint. Each unit is capable of representing a piece of the moving object by a "simple" function. Simple functions are lin-



Moving Object Languages, Figure 2 Structure of the type system



Moving Object Languages, Figure 3 Sliced representation of a *moving*(*real*) and a *moving*(*points*) value

ear functions for moving points or regions, and quadratic polynomials (or square roots of such) for moving reals, for example.

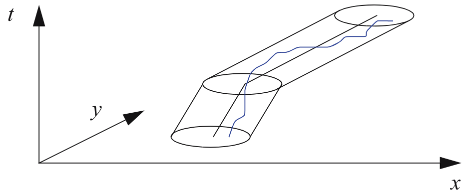
Within a database system, an extension module (data blade, cartridge, extender, etc.) can be provided offering implementations of such types and operations. The sliced representation is basically stored in an array of units.[3] Because values of moving object types can be large and complex, the DBMS must provide suitable storage techniques for managing large objects. A large part of this design has been implemented prototypically in the SECONDO extensible DBMS [1] which is available for download [12].

Some Further Work on Moving Object Languages

Moving Objects in Networks The model of spatio-temporal data types presented in the previous section has been extended to model movement in networks [8]. A network is modeled as a set of routes and junctions between routes. Four data types *gpoint*, *gline*, *moving*(*gpoint*) and *moving*(*gline*) are introduced to represent static and moving network positions and regions, respectively. Representative entities on a highway network would be gas stations, construction areas, vehicles, or traffic jams, for example. Some advantages over a model with free movement are that descriptions of moving objects become much more compact (as they do not contain geometries any more) such that relationships between moving objects and the underlying network can be easily used in queries, and that connectivity of the network is considered, e. g., for network distance or shortest path computations.

Spatio-temporal Predicates and Developments Erwig and Schneider [3,4] extend the spatio-temporal data type approach by considering developments of topological relationships over time. They develop a language to describe such developments in predicates that can then be used for filter and join conditions on moving objects. For example, consider an airplane traversing a storm area. The topological relationship between the moving point and the moving region will be *disjoint* for a period of time, then *meet* for an instant, then *inside* for a time interval, then *meet* again and finally *disjoint* again. The framework first allows one to obtain basic spatio-temporal predicates by aggregating a static topological relationship over all instants of a time interval. This is essentially done by lifting (as explained above) and existential or universal quantification. Exist-

[3] It is a bit more complicated in the case of variable size units as for a moving region, for example.



Moving Object Languages, Figure 4 Geometry of an uncertain trajectory

ing predicates can then be sequentially composed to derive new predicates. For example, we may define a predicate **Cross** to describe a development like the passing of the air plane through the storm as:

$$\textbf{Cross} := \textbf{Disjoint} \triangleright \textbf{meet} \triangleright \textbf{Inside} \triangleright \textbf{meet} \triangleright \textbf{Disjoint}$$

Uncertain Trajectories A *moving point* (Fig. 1) in the 2D + time space is in the literature often called a *trajectory*. If one represents it discretely as a polyline and takes an uncertainty threshold into account, geometrically it will obtain the shape of a kind of slanted cylinder (Fig. 4).

It is only known that the real position is somewhere inside this volume. Based on this model, Trajcevski et al. [14] have defined a set of predicates between a trajectory and a region in space taking uncertainty and aggregation over time into account, namely

PossiblySometimeInside	**SometimePossiblyInside**
PossiblyAlwaysInside	**AlwaysPossiblyInside**
DefinitelySometimeInside	**SometimeDefinitelyInside**
DefinitelyAlwaysInside	**AlwaysDefinitelyInside**

They also give algorithms for evaluating such predicates.

A text book covering the topics presented in this article in more detail is available [10].

Key Applications

Query languages of the first kind, that is, for querying current and near future movement like the MOST model described, are the foundation for location-based services. Service providers can keep track of the positions of mobile users and notify them of upcoming service offers even some time ahead. For example, gas stations, hotels, shopping centres, sightseeing spots, or hospitals in case of an emergency might be interesting services for car travelers.

Several applications need to keep track of the current positions of a large collection of moving objects, for example, logistics companies, parcel delivery services, taxi fleet management, public transport systems, air traffic control. Marine mammals or other animals are traced in biological applications. Obviously, the military is also interested in keeping track of fighting units in battlefield management.

Query languages of the second kind – for querying history of movement – are needed for more complex analyses of recorded movements. For example, in air traffic control one may go back in time to any particular instant or period to analyse dangerous situations or even accidents. Logistics companies may analyze the paths taken by their delivery vehicles to determine whether optimizations are possible. Public transport systems in a city may be analyzed to understand reachability of any place in the city at different periods of the day. Movements of animals may be analyzed in biological studies. Historical modeling may represent movements of people or tribes and actually animate and query such movements over the centuries.

Query languages of the second kind not only support moving point entities but also moving regions. Hence also developments of areas on the surface of the earth may be modeled and analyzed like the deforestation of the Amazone rain forest, the Ozone hole, development of forest fires or oil spills over time, and so forth.

Future Directions

Recent research in databases has often addressed specific query types like continuous range queries or nearest neighbour queries, and then focused on designing efficient algorithms for them. An integration of the many specific query types into complete language designs as presented in this entry is still lacking. Uncertainty may be treated more completely also in the approaches for querying history of movement. A seemless query language for querying past, present, and near future would also be desirable.

Cross References

► Spatio-temporal Data Types
► Trajectory

Recommended Reading

1. Almeida, V.T., Güting, R.H., Behr, T.: Querying moving objects in SECONDO. In: Proceedings of the Mobile Data Management Conference, Nara, Japan, pp. 47–51 (2006)
2. Cotelo Lema, J.A., Forlizzi, L., Güting, R.H., Nardelli, E., Schneider, M.: Algorithms for moving object databases. Comp. J. **46**(6), 680–712 (2003)
3. Erwig, M., Schneider, M.: Developments in spatio-temporal query languages. IEEE Int. Workshop on Spatio-temporal Data Models and Languages (STDML), pp. 441–449 (1999)
4. Erwig, M., Schneider, M.: Spatio-temporal predicates. IEEE Trans. Knowl. Data Eng. (TKDE) **14**(4), 881–901 (2002)
5. Erwig, M., Güting, R.H., Schneider, M., Vazirgiannis, M.: Spatio-temporal data types: an approach to modeling and querying moving objects in databases. GeoInformatica **3**, 265–291 (1999)

M

6. Forlizzi, L., Güting, R.H., Nardelli, E., Schneider, M.: A data model and data structures for moving objects databases. Proc. ACM SIGMOD Conf. Dallas, Texas, USA, pp. 319–330 (2000)
7. Grumbach, S., Rigaux, P., Segoufin, L.: The DEDALE system for complex spatial queries. Proc. ACM SIGMOD Conference, Seattle, Washington, pp. 213–224 (1998)
8. Güting, R.H., de Almeida, V.T., Ding, Z.: Modeling and querying moving objects in networks. VLDB Journal **15**(2), 165–190 (2006)
9. Güting, R.H., Böhlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N.A., Schneider, M., Vazirgiannis, M.: A foundation for representing and querying moving objects in databases. ACM Trans. Database Syst. **25**, 1–42 (2000)
10. Güting, R.H., Schneider, M.: Moving Objects Databases. Morgan Kaufmann, Amsterdam (2005)
11. Rigaux, P., Scholl, M., Segoufin, L., Grumbach, S.: Building a constraint-based spatial database system: model, languages, and implementation. Inform. Syst. **28**(6), 563–595 (2003)
12. SECONDO System. Available for download at http://www.informatik.fernuni-hagen.de/import/pi4/Secondo.html/
13. Sistla, A.P., Wolfson, O., Chamberlain, S., Dao, S.: Modeling and querying moving objects. Proc. 13th Int. Conf. on Data Engineering, Birmingham, UK, pp. 422–432 (1997)
14. Trajcevski, G., Wolfson, O., Hinrichs K., and Chamberlain, S.: Managing uncertainty in moving objects databases. ACM Trans. Database Syst. (TODS) **29**(3), 463–507 (2004)
15. Wolfson, O., Chamberlain, S., Dao, S., Jiang, L., Mendez, G.: Cost and imprecision in modeling the position of moving objects. Proc. 14th Int. Conf. on Data Engineering, Orlando, Florida, pp. 588–596 (1998)
16. Wolfson, O., Sistla, A.P., Chamberlain, S., Yesha, Y.: Updating and querying databases that track mobile units. Distrib. Parallel Database **7**, 257–387 (1999)
17. Wolfson, O., Xu, B., Chamberlain, S., Jiang, L.: Moving objects databases: issues and solutions. Proc. 10th Int. Conf. on Scientific and Statistical Database Management, Capri, Italy, pp. 111–122 (1998)

Moving Object Uncertainty[1]

GOCE TRAJCEVSKI, PETER SCHEUERMANN
Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA

Synonyms

Spatio-temporal uncertainty; GPS; Motion tracking; Location-based services; Moving object databases; Dead-reckoning

Definition

Uncertainty is an inherent component of any system that manages the data pertaining to the location-in-time information of mobile entities. Typically, such systems receive some form of an on-line *(location, time)* updates for individual moving objects, which may be obtained either from a GPS device on-board each moving object or by using some other tracking technology (e. g., PCS triangulation networks and motion tracking sensor networks). This information is transmitted to a repository which stores the data and can be used for providing answers to various queries of interest. However, there are several constraints which limit the accuracy of these data:

(1.) Due to bandwidth limitations, networks' connectivity and various clock-synchronization issues, the *actual* time that a given object was at a particular location may not be equal to the time that its presence at that location is recorded in the database. This, in a sense, resembles the traditional distinction between *valid time* and *transaction time* as studied in temporal databases [1].

(2.) The devices that are used to determine the location of a given object are prone to measurement-errors themselves. For example, depending on the number of satellites whose signals are available in a given region, the GPS-based error may range from few decimeters to a few meters [17]; the sensor nodes' coverage may not be sufficient to guarantee exact location [4].

(3.) Furthermore, one cannot make any exact claims about the location of a given object in-between consecutive updates [16,27].

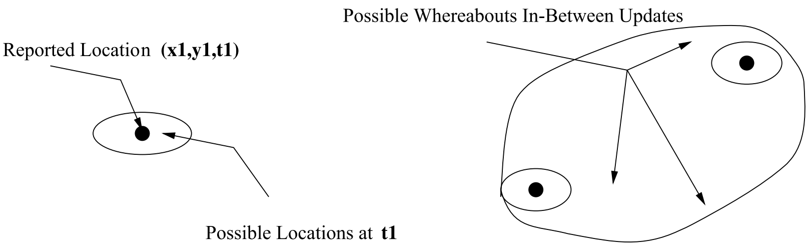
These basic concepts are illustrated in Fig. 1, where the left portion indicates the uncertainty with respect to the object's location for a particular update, and the right portion indicates the boundaries of the region of possible whereabouts of the object between the updates.

Historical Background

Miniaturization of computing devices and the advances of wireless communications and sensor technologies have spurred many research and implementation efforts into several classes of applications that can be grouped as Location-Based Services (LBS) [21]. An important aspect of almost any LBS system is the management (i. e., modelling, storing/retrieving and querying) of the transient location-in-time data pertaining to the mobile entities involved. As a result, in recent years, a large body of research works have emerged, which are collectively forming the field of Moving Objects Databases (MOD)[12].

Historically, researchers have independently pursued two complementary tracks that have extended the traditional database research. On one hand, the transactional aspects of database management systems were extended with temporal awareness and the field of temporal databases investigated various impacts that the semantics of time has on the

[1]Research Supported by the Northrop-Grumman Corp. (P.O.8200082518) and the NSF (IIS-0325144)



Moving Object Uncertainty, Figure 1
Example of moving object uncertainty

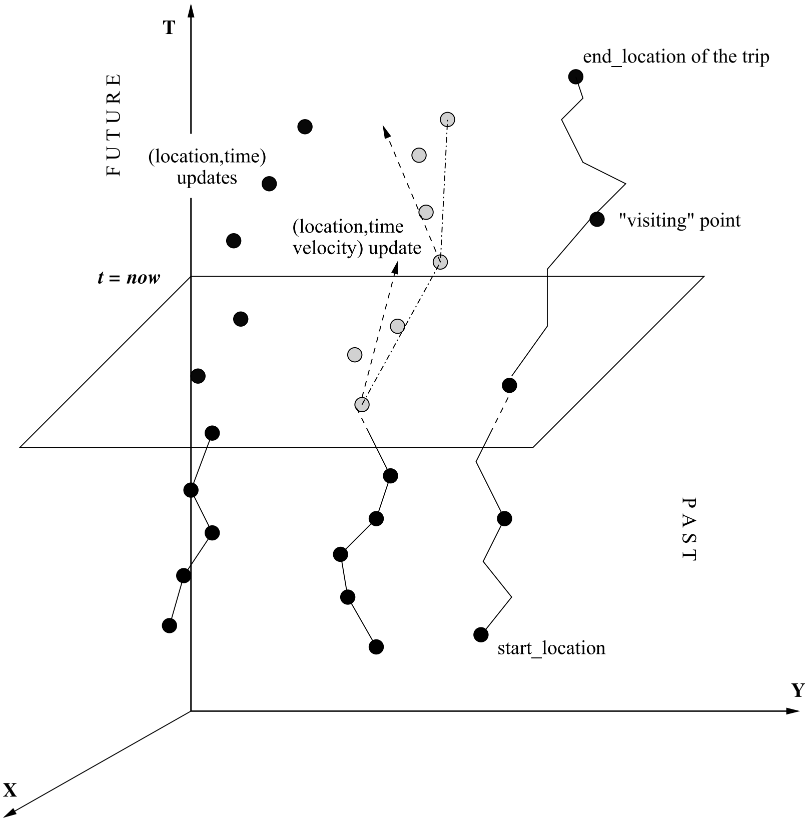
quality of the stored data with respect to the actual world being modelled [1]. On the other hand, the management (i. e., representation, indexing, querying) of the spatial data for applications such as GIS, in which the entities of interest are objects with particular geographic and dimensionality attributes (e. g., location, shape, extent) was investigated [20]. One of the first commercially-important applications in which the efficient management of the *(location, time)* information for a large number of mobile users is a paramount, was the cellular telephony. Various data management architectures were proposed for the efficient tracking, call-forwarding and billing of users [18]. Around the same time, motivated by various LBS applications, researchers recognized the need for more thorough treatment of spatio-temporal data, i. e., data whose spatial attributes (e. g., location) change over time. In particular [22], introduced the concept of dynamic attributes and spurred the development of the field of moving objects databases [12]. Due to the dynamic nature of the entities involved, data management in MOD settings has a number of distinguishing features, with respect to the traditional data management aspects of modelling/representing the data [11], indexing structures for accessing the data items [15,24], and algorithms and methodologies for processing the spatio-temporal queries [7,10,12]. In particular, the most widely used queries (range, (k)nearest-neighbor, join) become *continuous*. In other words, their answers change over time due to the changes in the locations of the moving objects, and efficient techniques are needed for maintaining the correctness of the answers.

Different works have used different models to represent the motion plans of the moving objects and, based on the model chosen, a plethora of indexing structures and algorithms for processing the popular categories of spatio-temporal queries have been proposed [12,24]. However, irrespective of the chosen representation, the *uncertainty* of the moving objects remains an inherent component. As has been pointed out in many works (e. g., [27,29]), unless the uncertainty is captured in the model itself, the burden of factoring it out from the meaning/answers of the various spatio-temporal queries will fall on the end user. Hence, we need the following: (1.) Models of the uncertainty as part of the objects' motion model. (2.) Linguistic constructs that will enable the users to specify queries in the presence of uncertainty. (3.) Techniques for efficient query processing for uncertain moving objects' data.

Scientific Fundamentals

Traditionally, there are three main models for representing the future motion plans of moving point objects. As a consequence of the chosen model, the researchers have developed corresponding algorithms for the processing of continuous spatio-temporal queries.

- At one extreme is the model in which the objects periodically send their *(location, time)* updates to the MOD server (left-most part in Fig. 2). Due to the frequency of the updates, intelligent methodologies are needed that will avoid constant re-evaluation of the pending continuous queries, while still ensuring the correctness of their answers [14]. In order to balance the efficiency of query processing with keeping the MOD up-to-date, recent works have also addressed "lazy" updating mechanisms [30].
- In the "middle-land" is the model in which the moving objects are assumed to periodically send *(location, time, velocity)* updates to the MOD server [22], as illustrated by the middle portion of Fig. 2. Efficient algorithms for processing continuous queries in MOD under this model were presented in [13], and [10] addressed the distributed processing of such queries, by delegating some of the responsibilities to the moving objects themselves. One peculiar feature of this model is that in-between two consecutive updates, the objects are allowed to deviate from the expected route which is calculated using the *velocity* parameter of the most recent update, for as long as the deviation is within certain tolerance bounds [29]. This is illustrated by the shaded circles in the middle portion of Fig. 2, which shows the actual locations-in-time, as opposed to the expected ones that would be along the dotted arrowed line.
- The other extreme model is the one in which the entire future motion plan of a given object is represented as a *trajectory* (right-most portion in Fig. 2). Under this model, each object is assumed to initially transmit to the MOD server the information about its *start_location*,



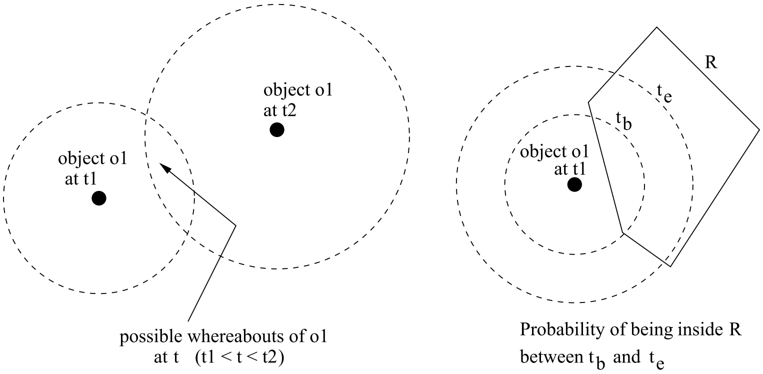
Moving Object Uncertainty, Figure 2
Modelling the motion plans of moving objects

end_location, and *start_time* of the trip, plus (possibly) a set of "to-be-visited" points. Using the information available from the electronic maps, plus the knowledge about the distribution of the traffic patterns in a given time-period, the server will apply an A^*-like variant of the time-aware Dijkstra's algorithm to generate the optimal travel plan [27]. A trajectory is essentially a sequence of 3D points (2D geography + time) of the form $(x_1, y_1, t_1), (x_2, y_2, t_2), \ldots, (x_n, y_n, t_n)$, where $t_1 < t_2 < \ldots t_n$ and in-between two points the object is assumed to move along a straight line and with a constant speed. The peculiarity of this model is that it enables answering continuous queries pertaining to the further-future; however, the consequence is that a disturbance of the traffic patterns in a small geographic region may affect the correctness of the queries in widely-dispersed areas and at different time-intervals [7].

An important observation is that when it comes to the *past* portion of the objects' motion, all three models, in a sense, converge, and represent it as a trajectory (bottom part of Fig. 2).

Any model of the uncertainty of the moving objects, as well as its implications on the query processing algorithms, is closely associated with the adopted model for representing the objects' motion plans.

As we already have illustrated in Fig. 1, even a single location update at a given time-instance is associated with its own uncertainty. Furthermore, if the motion plan of the moving objects is represented as a sequence of *(location, time)* updates, then there are other consequences of the uncertainty. As an illustration, observe the left portion of Fig. 3. Assume that it represents a scenario in which an *exact* value of of the locations of a given object is known at times t_1 and t_2 (also assuming that the consecutive updates were sent at those times). However, even under this assumption (and, again, ignoring the fact that these very values already have uncertainty associated with them, c.f, Fig. 1.), there is still some imprecision regarding the object's motion. Namely, given only the velocity bounds of the object o_1, its location at a time t ($t \in (t_1, t_2)$) cannot be exactly determined. As illustrated in the left portion of Fig. 3 (assuming that $(t - t_1) < (t_2 - t)$), at t, o_1 can be anywhere inside the lens obtained as the intersection of the of the two circles bounding the object's possible locations with respect to its maximum speed limit. It can be demonstrated that all the possible whereabouts of the object for the time-interval (t_1, t_2) are bound by an ellipse with foci at the locations reported at the respective times [17]. If one assumes a certain probability distribution corresponding to the objects uncertainty [4], then algo-



Moving Object Uncertainty, Figure 3 Uncertainty for the (location, time) model

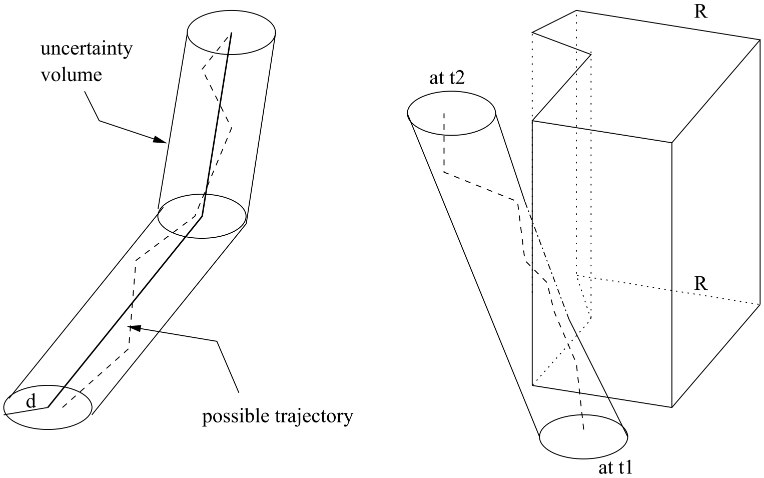
rithms can be developed for answering the spatio-temporal queries with uncertainty. As a particular example, illustrated by the right portion of Fig. 3, one can pose the range query: **QR1:** *"What is the probability that the object* o_1 *will be inside the region R between* t_b *and* t_e*"*, for which the value is obtained by calculating the area of the intersection of the circular ring bounding the location of the object between t_b and t_e, with the polygon bounding the region *R*, and dividing the result with the area of the circular ring. Another variant of the continuous range query with uncertainty is: **QR2:** *"Retrieve all the objects that have more than 75% chance of being inside the region R between* t_b *and* t_e*"*. One can easily envision the impact of the uncertainty on the other types of continuous spatio-temporal queries. For example, a variant of the nearest-neighbor query would be: **Q-NN:** *"Retrieve all the objects that have more then 90% chance of being nearest neighbors to the object* o_1 *between* t_b *and* t_e*"*.

When the motion plan of the moving object is modelled as a sequence of *(location, time, velocity)* updates (c.f., the middle portion of the Fig. 2), the uncertainty is already incorporated as a part of the *contract/agreement* between the MOD server and the individual mobile objects. Namely, in order to minimize the communication overhead, after each update, the particular mobile object knows what is the *expected* location that the server calculates based on the *velocity* parameter. For as long as the expected location does not deviate by more than a certain pre-defined threshold, say, ε, from the actual location of the object obtained by some measurement (e. g., an on-board GPS device), the object will not send new updates to the MOD server. This kind of update policy is known as a *dead-reckoning*, and different variants of it have been explored in detail in [29], along with the analytic expressions for deriving the overall *information cost* of keeping the (im)precision of the MOD data within desired bounds. Clearly, every spatio-temporal query under this model has an explicit uncertainty in its answer, due to the location error-bound of ε. Recent algorithms for efficient processing of the spatio-temporal queries under these settings were presented in [10].

For the case when the future motion plan of a moving object is represented as a *trajectory*, since one of the parameters used in its construction is the distribution of the speed-patterns on the road-segments, the uncertainty represents the acceptable deviation of the objects due to traffic fluctuations.

The implications of this assumption are illustrated in Fig. 4. Namely, if the object is expected to be at some location, say, (x_1, y_1) at a given time t_1, the *uncertainty area* of its whereabouts is a disk with a radius *d*, centered at the expected location (x_1, y_1). Extending this over a time-interval, the set of all the *possible trajectories* of a given moving object defines an *uncertainty volume*, which in 3D settings (2D geography + time) is represented as a sequence of sheared cylinders, one for each straight-line segment of the object's route. Detailed algorithms for processing continuous spatio-temporal queries under these settings were presented in [27], where the *qualitative* uncertainty of the queries was discussed. Namely, to test for the satisfiability of the spatio-temporal predicates, the quantifiers *possibly* and *definitely* were considered in the spatial dimension, and the quantifiers considered in the temporal dimension were *sometimes* and *always*. The query operators corresponded to the various predicates that can be composed by interleaving the order of the quantification. Thus, the right portion of Fig. 4 presents an example of an uncertain trajectory which satisfies the predicate: *"Possibly inside region R, sometimes between* t_1 *and* t_2*"*. An approach assuming quantitative probabilistic values for the answer to such queries, under the assumption of uniform distribution for the probability of the object being inside the disk with radius *d* was presented in [25].

Clearly, one cannot dwell on query processing for any large data sets unless there are proper indexing techniques for retrieving that data, especially during the filtering stage in which disk-accesses should be minimized, while ensuring as few false-positives as possible. There has been



Moving Object Uncertainty, Figure 4 Uncertainty for the (location, time) model

a plethora of indexing structures proposed for processing continuous spatio-temporal queries (e. g., see the references in [12]). Furthermore, recent research works have specifically focused on developing efficient indexes which explicitly take into consideration the uncertainty of the mobile entities [6]. One of the main benefits of incorporating the uncertainty into the index is that certain objects can be pruned from the search earlier during the filtering part of the processing of probabilistic queries.

Some of the recent works targeting efficient management of continuous spatio-temporal queries for moving objects have explicitly focused on deeper exploitation of the fact that in many applications of interest, the objects are moving on road networks [9]. This additional semantic knowledge can be exploited to obtain further gains in the efficiency of objects' tracking and query processing. Recently, techniques have been developed for efficient tracking and indexing of moving objects on road networks [6] and, in particular, the impact of uncertainty under such settings has been investigated in [8].

Key Applications

Moving object uncertainty is of interest in several scientific and application domains.

GIS

Typically, in MOD research, the moving objects are approximated as points whose dimensions (e. g., size) can be neglected with respect to the overall area/volume of the universe of discourse. However, many of the objects of interest in GIS can not be approximated as points – namely, rivers have their shapes, forests cover areas typically represented as polygon-bounded regions, fires are spreading in time and are represented as moving polygons [11]. However, the boundaries between regions are seldom precise (e. g., a boundary between a prairie and a desert). Hence, one must account for the uncertainty in the representation and use the corresponding mathematical tools to model it and develop query algorithms (e. g., fuzzy-set theory [17]).

LBS

A variety of applications in LBS can use the uncertainty inherent to mobile objects. As a typical example, in tourist-information systems, the main concern is how to provide a context-aware delivery of the data which matches the preferences of a given user based on its location [21]. However, if revealing the exact location of the mobile user can be adversely used for violating some of the privacy issues, uncertainty can be used to provide some forms of location-based privacy. As another example, depending on the trade-offs between the desired uncertainty of the information (e. g., as a function of the distance of a given moving object from a given target) and the size of data kept in the cache, methodologies have been devised which implement various policies for keeping/purging data items in/out a given cache [5].

Meteorology and Seismology

Many of the phenomena of interest in meteorology are entities which move and even change their shape over time (e. g., clouds, flood regions). Clearly, one cannot exactly model their shapes and bounding regions/volumes and for the purpose of query processing and any kind of reasoning involved, the uncertainty must be incorporated as part of the representation and the evolution of the objects [19]. Furthermore, the probabilistic values due to uncertainty must be properly taken into consideration if the system, which monitors the phenomena of interest, is expected to exhibit some form of reactive behavior [17,26]. These issues are extremely important when, based on the changes of the monitored values (e. g., the coastal erosion co-related with seismic measurements in tsunami prediction; the CO-concentration co-related sulphur concentration and the temperature increases for predicting volcano's eruption) disaster-preventing alarms need to be issued with certain probabilistic guarantees [23].

Bio-Chemistry

The efficient management of a vast body of observational data generated by expensive experiments is a paramount for researchers in biology and chemistry. As a particular example, the modelling of the information representing large sequences of metabolic pathways requires special tools to store, visualize and query such data. Due to the inherent properties of the micro-world, uncertainty is a natural parameter in such data sets. However, an important part of the research in biology and chemistry is related to modelling and reasoning about the reactions that may occur in various experiments. In such settings, the micro-mobility of the compounds which bind themselves in larger structures during the process of a given experiment cannot be specified in a crisp manner. Consequently, some researchers have recently incorporated the uncertainty when modelling the dynamics of the metabolic control [28].

Sensor Networks

The management of spatio-temporal data in sensor networks is a relatively new field with many open challenges and its natural settings simply cannot avoid the uncertainty. Namely, regardless of what kind of sensors are used for tracking mobile objects, the precision of determining their location is limited. Furthermore, due to the discrete coverage of a given geographic region of their deployment, determining the set of sensors that should process a particular query (e. g., the boundary of a given region for a range query) introduces yet another source of uncertainty [2]. Yet another domain-specific problem that arises in these settings, and is a natural source for the imprecision of the data, is due to the fact that the quality of the readings of the sensor nodes are (an inverse) function of the mobile object's distance and, moreover, maintaining the identities of the individual objects (for the purpose of correct answer to the queries) is a challenge of its own [31].

Spatio-temporal Data Reduction and Data Mining

The goal of any data reduction method is to decrease the size of the data-set of interest for a particular application. Clearly, there is a trade-off between the amount of data reduced and the level of the uncertainty introduced (with respect to the original data set). As demonstrated in [3], unless proper caution is exercised when selecting the distance-function used in the reduction process, the errors obtained for the widely used class of spatio-temporal queries may become unbounded. Similar trade-offs, involving the uncertainty as part of the model, are present when, for various data mining purposes, one needs to cluster a set of trajectories representing the typical motions of moving objects along given routes and within given time-intervals and perform some similarity-based reasoning.

Future Directions

In general, any system which is targeted towards managing the *(location, time)* information pertaining to large amounts of mobile entities, is bound to incorporate uncertainty into its model and, as a consequence, in the processing algorithms for users' queries. One of the applications that poses a very intriguing challenge is the field of mobile data management in sensor network settings. In particular, the uncertainty present in these settings has a variety of sources, e. g., imprecision of the devices used (as a function of the object's distance), impossibility of the perfect coverage of the regions of interest, etc. However, this uncertainty can be exploited for the purpose of optimizing the in-network processing of various queries, in the sense that one can minimize the transmission of the individual sampling results when they are within the predefined bounds. This way, one can reduce the consumption of the most expensive resource – the energy of the individual nodes, which is mostly consumed when data transmission is required.

As pointed out in [17], a thorough treatment of uncertainty, besides the adopted model used for its representation, needs a solid apparatus for executing the operations over the data. One of the challenges of managing the mobile object uncertainty is finding a proper fusion of probability theory, fuzzy-sets theory and other existing fields (e. g., computational geometry [2]) that will best serve the needs of a given application domain.

Cross References

► Indexing Schemes for Multi-dimensional Moving Objects
► Privacy Threats in Location-Based Services
► Spatio-temporal Query Languages
► Uncertainty, Modeling with Spatial and Temporal

M

References

1. Böhlen, M.H., Jensen, C.S.: Temporal data model and query language concepts. Encyclopedia of Information Systems, 4, Academic Press, Inc. (2003)
2. Buragohain, C., Gandhi, S., Hershberger, J., Suri, S.: Contour approximation in sensor networks. In: DCOSS (2006)
3. Cao, H., Wolfson, O., Trajcevski, G.: Spatio-temporal data reduction with deterministic error bounds. Journal of Very Large Databases **15**(3) (2006)
4. Cheng, R., Kalashnikov, D.V., Prabhakar S.: Querying imprecise data in moving object environments. IEEE Transactions on Knowledge and Data Engineering. **16**(9) (2004)
5. Cherniack, M., Galvez, E.F., Franklin, M., Zdonik, S.: Profile-driven cache management. In: ICDE (2003)
6. de Almeida, V.T., Güting, R.H.: Indexing the trajectories of moving objects in networks. GeoInformatica, **9**(1) (2005).
7. Ding, H., Trajcevski, G., Scheuermann, P.: Efficient maintenance of continuous queries for trajectories. GeoInformatica (2007). doi: 10.1007/s10707-007-0029-9
8. Ding, Z. and Güting, R.H.: Uncertainty management for network constrained moving objects. In: DEXA (2004)
9. Ding, Z. and Güting, R.H.: Managing moving objects on dynamic transportation networks. In: International Conference on Scientific and Statistical Database Management (SSDBM) (2004)
10. Gedik, B. Liu, L.: Mobieyes: A distributed location monitoring service using moving location queries. IEEE Transactions on Mobile Computing. **5**(10) (2006)
11. Güting, R.H., Bohlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N., Nardeli, E., Schneider, M., Viqueira, J.R.R.: Spatio-temporal models and languages: An approach based on data types. In: Spatio-temporal Databases – the Chorochronos Approach. (2003)
12. Güting, R.H. Schneider, M.: Moving Objects Databases. Morgan Kaufmann, San Francisco, CA (2005)
13. Iwerks, G.S., Samet, H., Smith, K.P.: Maintenance of k-nn and spatial join queries on continuously moving points. ACM Transactions on Database Systems. **31**(2) (2006)
14. Mokbel, M.F., Xiong, X., Aref, W.G.: Sina: Scalable incremental processing of continuous queries in spatio-temporal databases. In: ACM SIGMOD Internation Conference on Management of Data (2004)
15. Pelanis, M., Saltenis, S., Jensen, C.S.: Indexing the past, present, and anticipated future positions of moving objects. ACM Transactions on Database Systems. **31**(1), (2006)
16. Pfoser D., Jensen, C.: Capturing the uncertainty of moving objects representation. In: SSDB (1999)
17. Pfoser D., Tyfona, N., Jensen, C.: Indeterminacy and Spatiotemporal Data: basic definitions and case study. Geoinformatica. **9**(3) (2005)
18. Pitoura E., Samaras, G.: Locating objects in mobile computing. IEEE Transactions on Knowledge and Data Engineering. **13**(4) (2001)
19. Reiners, W.A.: Transport of energy, information and material through the biosphere. Annual Review of Environment and Resources. **28**(1) (2003)
20. Samet, H.: Foundations of Multidimensional and Metric Data Structures. Morgan Kaufmann, San Francisco, CA (2006)
21. Schiller, J., Voisard, A.: Location-based Services. Morgan Kaufmann, San Francisco, CA (2004)
22. Sistla, A.P., Wolfson, O., Chamberlain, S., Dao, S.: Modeling and querying moving objects. In: 13th Int'l Conf. on Data Engineering (ICDE) (1997)
23. Stora, D., Agliati, P., Cani, M., Neyret, R., Gascuel, J.: Animating lava flows. In: Graphics Interfaces (1999)
24. Tao, Y., Papadias, D., Sun., J.: The tpr*-tree: An optimized spatio-temporal access method for predictive queries. In: VLDB (2003)
25. Trajcevski, G.: Probabilistic range queries in moving objects databases with uncertainty. In: MobiDE (2003)
26. Trajcevski, G., Scheuermann, P., Ghica, O., Hinze, A., Voisard, A.: Evolving triggers for dynamic environments. In: Extending Database Technology (EDBT) (2006)
27. Trajcevski, G., Wolfson, O., Hinrichs, K., Chamberlain, S.: Managing uncertainty in moving objects databases. ACM Transactions on Database Systems. **29**(3) (2004)
28. Wang, L., Birol, I., Hatzimanikatis, V.: Metabolic control analysis under uncertainty: framework development and case studies. Biophysical Journal. **87**(6) (2004)
29. Wolfson, O., Sistla, A.P., Chamberlain, S., Yesha, Y.: Updating and querying databases that track mobile units. Distributed and Parallel Databases, 7 (1999)
30. Xing, X., Aref, W.: R-trees with update memos. In: ICDE (2006)
31. Zhao, F., Guibas, L.: Wireless Sensor Networks: an Information Processing Approach. Morgan Kaufmann, San Francisco, CA (2004)

Moving Objects

▶ Indexing of Moving Objects, B^x-Tree
▶ Indexing Schemes for Multi-dimensional Moving Objects
▶ Indexing Spatio-temporal Archives

Moving Objects Database

▶ Mobile Objects Databases

Moving Points

▶ Constraint Databases and Moving Objects
▶ Trajectory

Moving Queries

▶ Continuous Queries in Spatio-temporal Databases

Moving Regions

▶ Constraint Databases and Moving Objects

MRA-Tree

▶ Multi-Resolution Aggregate Tree

M&S Computing

► Intergraph: Real Time Operational Geospatial Applications

M-Tree

► Indexing, High Dimensional

Multi Agent Systems

► Wayfinding: Affordances and Agent Simulation

Multicriteria Decision Making, Spatial

SALEM CHAKHAR, VINCENT MOUSSEAU
LAMSADE, University of Paris Dauphine, Paris, France

Synonyms

Spatial multicriteria decision aid; GIS-based multicriteria decision analysis; Decision-making, multi-criteria; Decision-making, multi-attribute; Decision-making, multi-objective; Decision rules; Analysis, sensitivity; Analysis, robustness; Preference structure; Mathematical programming

Definition

Multicriteria analysis is generally defined as "*a decision-aid and a mathematical tool allowing the comparison of different alternatives or scenarios according to many criteria, often conflicting, in order to guide the decision maker toward a judicious choice*" [12]. The set of decision alternatives considered in a given problem is often denoted by A and called the *set of potential alternatives*. A *criterion* is a function g, defined on A, taking its values in an ordered set and representing the decision maker's preferences according to some points of view. The evaluation of an alternative a according to criterion g is written $g(a)$.

Spatial multicriteria decision making refers to the application of multicriteria analysis in spatial contexts where alternatives, criteria and other elements of the decision problem have explicit spatial dimensions. Since the late 1980s, multicriteria analysis has been coupled with *geographical information systems* (GIS) to enhance spatial multicriteria decision making.

Historical Background

It is generally assumed that multicriteria analysis was born and took its actual vocabulary and form at the beginning of 1960s. In fact, most multicriteria analysis practitioners consider that their field stems largely from the research of Simon on satisficing and the early works on goal programming. Closely related to decision-making in general and to multicriteria analysis in particular is utility theory. Although utility theory was first used to model simple individual preferences, it has been extended to multicriteria preferences and led to the *multiattribute utility theory* [7].

The first methods in multicriteria analysis were developed during the 1960s. Goal programming, for example, uses linear programming method to resolve a multicriteria problem. In 1968, Roy conceived the initial version of the ELECTRE method (see [4]).

Throughout the 1970s, the widely dispersed scientific field of multicriteria analysis started to take form. First, in 1971 Roy organized the first independent session specifically devoted to multicriteria research within the 7th Mathematical Programming Symposium, held in The Hague. Second, in 1972 Cochrane and Zeleny organized the First International Conference on multicriteria decision making at the University of South Carolina. Then in 1975, Roy organized the first meeting of the EURO Working Group on Multi-Criteria Decision Aid in Brussels. Also in 1975, Thiriez and Zionts organized the First Conference of the International Society on multicriteria analysis. In addition to these first scientific meetings, multicriteria analysis research focused in the 1970s on the theoretical foundations of *multiobjective decision making*.

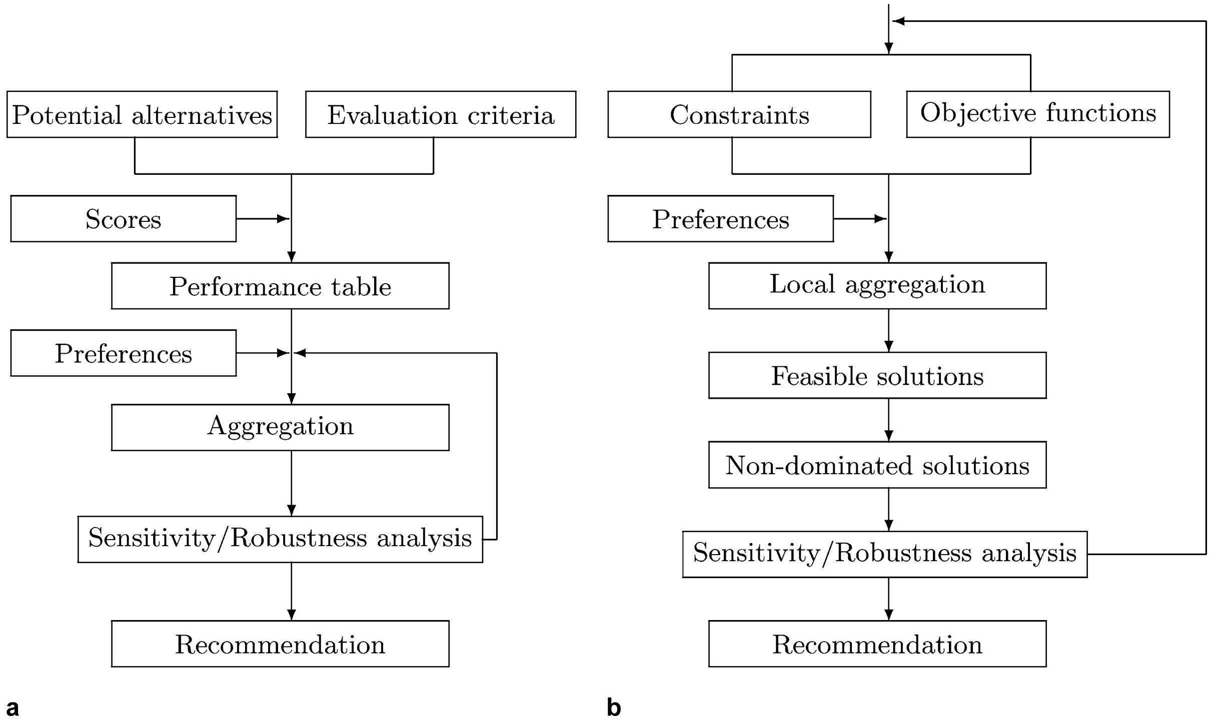
The 1980s and 1990s witnessed the consolidation and development of a great number of interactive methods. Most of these methods are oriented toward negotiation or multiple decision makers and multicriteria decision support systems.

Multicriteria analysis has been used since its emergence to deal with spatial decision problems. The first works involving GIS-based multicriteria analysis where published in the late 1980s and the early 1990s. Currently, there are a number of relatively important articles devoted to GIS-based multicriteria analysis that have been published [10].

Scientific Fundamentals

General Schema of Multicriteria Analysis Methods

Different multicriteria analysis methods are available in the literature [4]. An excellent online bibliography of multicriteria analysis and its applications is available at http://www.lamsade.dauphine.fr/mcda/biblio/. Multicrite-



Multicriteria Decision Making, Spatial, Figure 1 General schema of discrete **a** and continuous **b** multicriteria methods

ria methods are commonly categorized as *discrete* or *continuous*, depending on the domain of alternatives. The former deals with a discrete, usually limited, number of pre-specified alternatives. The latter deals with variable decision values to be determined in a continuous or integer domain of infinite or large number of choices. Several authors classify them as (i) *multiple attribute decision-making* (MADM), and (ii) *multiple objective decision-making* (MODM). In this presentation, the discrete/continuous classification is chosen since it is in accordance with the conventional representation of data in GIS (vector vs. raster) and it is more general than the MADM/MODM classification. Figure 1 gives the general schema of discrete and continuous multicriteria methods that will be briefly described in the following two paragraphs.

Discrete Methods The first requirement of nearly all discrete techniques is a *performance table* containing the evaluations or *criteria scores* of a set of alternatives on the basis of a set of criteria. The next step consists of the aggregation of the different criteria scores using a specific *decision rule* (or *aggregation procedure*). It takes into account the *decision maker's preferences*, generally represented in terms of *weights* that are assigned to different criteria. The aggregation of criteria scores permits the decision maker to make a comparison between the different alternatives on the basis of these scores. The aggregation procedures represent the identities of the multicriteria analysis techniques. The discrete methods are usually categorized based on their aggregation procedures into two different families: (1) *outranking relation-based decision rules*, and (2) *utility function-based decision rules*.

The uncertainty and fuzziness generally associated with any decision situation require a *sensitivity/robustness analysis* enabling the decision maker(s) to test the consistency of a given decision or its variation in response to any modification in the input data and/or in the decision maker preferences.

Continuous Methods The starting point of most continuous methods are a set of *constraints* and *objective functions*. The former set contains inequalities which reflect natural or artificial restrictions on the values of the input data. This means that *feasible solutions* are *implicitly* defined in terms of these constraints.

For continuous methods, the decision maker's preferences generally take the form of *weights* that are assigned to different objective functions. They may also be represented as *target values* that should be satisfied with any feasible solution. The decision maker should also indicate, for each objective function, its *direction of optimization*,

that is maximization or minimization. No other information than the weights and these directions of optimization are required to define the set of *non-dominated solutions*. This set contains solutions that are not dominated by any other one.
Generally, *local* and *interactive* aggregation algorithms are used to define the feasible solutions set. This permits the combination of the decision maker preferences and the computer to solve the decision problem, using methods that alternate calculation steps and dialogue steps. In reality, the local and interactive algorithms require the decision maker preferences to be expressed *progressively* throughout the resolution process. The decision maker preferences, however, may be expressed *a priori* (i. e., before the resolution process) or *a posteriori* (i. e., after the resolution process).
In many practical situations, the decision maker is called upon to relax some of its constraints in order to guarantee that the set of feasible solutions is not empty or, simply, to test the stability of the results.

Spatial Multicriteria Decision Making

A brief description of spatial multicriteria decision making concepts is provided in the following. In the rest of this entry, $F = \{1,2,\cdots,m\}$ denotes the set of the indices of m evaluation criteria $g_1, g_2, \cdots, g_m$. Accordingly, g_j ($j \in F$) is the evaluation criterion number j.

Spatial Decision Alternatives Decision alternatives can be defined as alternative courses of action among which the decision maker must choose. A spatial decision alternative consists of at least two elements [9]: *action* (what to do?) and *location* (where to do it?). The spatial component of a decision alternative can be specified *explicitly* or *implicitly* [10]. The second case holds when there is a spatial implication associated with implementing an alternative decision.
The set of spatial decision alternatives may be discrete or continuous. In the first case, the problem involves a discrete set of pre-defined decision alternatives. Spatial alternatives are then modeled through one or a combination of the basic spatial primitives, namely point, line, or polygon. The second case corresponds to a high or infinite number of decision alternatives, often defined in terms of constraints. For practical reasons, the set of potential alternatives is often represented in a "discretized" form where each raster represents an alternative. Alternatives may be constructed as a collection of rasters.

Evaluation Criteria In the spatial context, evaluation criteria are associated with geographical entities and relationships between entities, and can be represented in the form of maps. One should distinguish a simple map layer from a *criterion map*. In fact, a criterion map models the preferences of the decision maker concerning a particular concept, while a simple map layer is a representation of some spatial real data. A criterion map represents subjective preferential information. Two different persons may assign different values to the same mapping unit in a criterion map.

Constraints A *constraint* (or *admissibility criterion*) represents natural or artificial restrictions on the potential alternatives. Constraints are often used in the pre-analysis steps to divide alternatives into two categories: "*acceptable*" or "*unacceptable*". An alternative is acceptable if its performance on one or several criteria exceeds a minimum or does not exceed a maximum.
In practice, constraints are often modeled through elementary multicriteria methods like the *conjunctive* or *disjunctive* aggregation procedures. With the conjunctive method, a *minimal satisfaction level* $\hat{g}_j$ is associated with each criterion g_j. If the performance of an alternative with respect to different criteria is equal or better to these minimal satisfaction levels (i. e., $g_j(a_i) \geq \hat{g}_j, \forall j \in F$), the alternative is considered as acceptable. Otherwise, the alternative is considered as unacceptable. With the disjunctive method, the alternative is considered acceptable as soon as at least one satisfaction level is exceeded.

Quantification The evaluation of alternatives may be quantitative or qualitative. Several methods require quantitative evaluations. In the literature, there exist some totally qualitative methods such as the median ranking method. Other methods, such as the ELECTRE family of methods (see [4]), involve both types of evaluations. When most of the criteria are qualitative, quantitative criteria may be converted into qualitative ones and a qualitative method used. Otherwise, a *quantification method* (i. e., assignment of numeric values to qualitative data) is applied; the *scaling approach* is the one most used.
Application of a quantification method requires the definition of a measurement scale. The most used measurement scale is the *Likert-type*. This scale is composed of approximatively the same number of favorable and unfavorable levels. An example with five levels is: *very unfavorable*, *unfavorable*, *neutre*, *favorable*, *very favorable*. Other more detailed measurement scales may also be used. The quantification procedure consists of constructing a measurement scale like the one with five points mentioned above. Then, numerical values are associated with each level of the scale. For instance, the numbers 1, 2, 3, 4 or 5 may be associated with the five-point scale from *very unfavorable* to *very favorable*.

Standardization The evaluation of alternatives may be expressed according to different scales (ordinal, interval, ratio). However, a large number of multicriteria methods (including practically all the utility function-based methods) require that all the criteria are expressed in a similar scale. Standardizing the criteria permits the rescaling of all the evaluation dimensions between 0 and 1. This allows between and within criteria comparisons.

There are a large number of standardization procedures. In all procedures, standardization starts from an initial vector $(g_j(a_1), g_j(a_2), \cdots, g_j(a_m))$ to obtain a standardized vector $(r_{1j}, r_{2j}, \cdots, r_{mj})$ with $0 \le r_{ij} \le 1$; $\forall j \in F$ and $i = 1, \cdots, n$ (n is the number of alternatives). The most used standardization procedure in GIS-based multicriteria decision making is the *linear transformation procedure*. It associates with each alternative a_i and for each criterion g_j the percentage of the maximum over all alternatives:

$$r_{ij} = \frac{g_j(a_i)}{\max_i g_j(a_i)}, \quad i = 1, \ldots, n; \;\; j \in F.$$

Pre-Analysis of Dominance In the absence of any preferential information, the only possible operation on the performance table is to eliminate the dominated alternatives. Let a and b be two alternatives from A. The alternative *a dominates* the alternative b in respect to F, noted as $a\Delta b$, if and only if:

$$g_j(a) \ge g_j(b); \quad j \in F,$$

with at least one strict inequality. Then, an alternative a from A is said to be *efficient*, *admissible* or *Pareto optimal* if and only if there is no other alternative b in A such that: $b\Delta a$.

Criteria Weights Generally, in multicriteria problems the decision maker considers one criterion to be more important than another. This *relative importance* is usually expressed in terms of numbers, often called *weights*, which are assigned to different criteria. These weights deeply influence the final choice and may lead to a non-applicable decision mainly when the interpretations of such weights are misunderstood by the decision maker.

In the literature, many direct weighting techniques have been proposed. When a *simple arrangement technique* is used, the decision maker sets the criteria in an order of preference. The *cardinal simple arrangement technique* involves each criterion being evaluated according to a pre-established scale. Other indirect methods are also available such as the *interactive estimation method*. There are also relatively complex weight assignment techniques such as the *indifference trade-offs* technique [7] and the *analytic hierarchy process* (AHP) [13].

Preference Structure and Preference Parameters When comparing two alternatives a and b, the decision maker will generally have one of the three following reactions: (i) preference for one of the two alternatives, (ii) indifference between the two alternatives or (iii) impossibility to compare the alternatives. These situations are generally denoted as follows: (i) aPb if a is preferred to b (bPa if it is the opposite), (ii) aIb if there is indifference between a and b, and (iii) aRb if there is an incomparability. The binary relations of *preference P*, *indifference I*, and *incomparability R* are respectively the sets of tuples (a,b) such that aPb, aIb, aRb. It is generally admitted that I is reflexive and symmetric, P is asymmetric, and R is irreflexive and symmetric. The three relations (I,P,R) constitute a *structure of preference* over A if and only if they have the properties mentioned above and only one of the following situations holds [14]: aPb, bPa, aIb, aRb.

Preference models require the definition of one or several thresholds, called *preference parameters*. The most commonly used preference parameters are the *indifference*, *preference* and *veto* thresholds. These three parameters are used essentially within the outranking relation-based decision rules. The first two parameters are for modeling imprecision and uncertainty in the decision maker's preferences. The latter is often used to compute the *discordance index*.

Decision Rules To compare alternatives in A, it is necessary to aggregate the *partial evaluations* (i. e., with respect to each criterion) into a global one by using a given *decision rule* (or *aggregation procedure*). As mentioned earlier, within the discrete family, there are usually two aggregation approaches: (i) *utility function-based approach*, and (ii) *outranking relation-based approach*. The basic principle of the first family is that the decision maker looks to maximize a utility function $U(a) = U(g_1(a), g_2(a), \cdots, g_m(a))$, aggregating the partial evaluations of each alternative into a global one. The simplest and most often used utility function has an additive form: $U(a) = \Sigma_{j \in F} u_j(g_j(a))$; where u_j are the partial utility functions. Within this form, the preference P and indifference I binary relations are defined for two alternatives a and b as follows:

$$aPb \Leftrightarrow U(a) > U(b) \quad \text{and} \quad aIb \Leftrightarrow U(a) = U(b).$$

In contrast with the first family, the second one uses *partial aggregation procedures*. Different criteria are aggregated into a partial binary relation S, with aSb used to indicate that "*a is at least as good as b*". The binary relation S is called an *outranking relation*. The most well known method in this family is ELECTRE (see, e. g., [4]). To construct the outranking relation S, for each pair of alternatives

(a, b), a *concordance index* $C(a, b) \in [0,1]$ – measuring the power of criteria that are in favor of the assertion aSb – and a *discordance index* $ND(a, b) \in [0,1]$ – measuring the power of criteria that are opposed to aSb – are computed. Then, the relation S is defined as follows:

$$\begin{cases} C(a,b) \geq \hat{c} \\ ND(a,b) < q\hat{d} \end{cases}$$

where $\hat{c}$ and $\hat{d}$ are the *concordance* and the *discordance thresholds*, respectively. Often an exploitation phase is needed to extract information from S on how alternatives compare to each other. At this phase, the concordance $C(a, b)$ and discordance $ND(a, b)$ indices are used to construct an index $\sigma(a, b) \in [0,1]$, representing the *credibility* of the proposition aSb, $\forall\,(a,b) \in A \times A$. The proposition aSb holds if $\sigma(a,b)$ is greater or equal to a given *cutting level*, $\lambda \in [0.5,1]$.

In the continuous formulation of a multicriteria problem, decision rules implicitly define the set of alternatives in terms of a set of *objective functions* and a set of *constraints* imposed on the decision variables. Here, *multiobjective mathematical programming* is often used. A multiobjective mathematical program is a problem where the aim is to find a vector $\mathbf{x} \in \mathbf{R}^p$ satisfying constraints of type

$$h_i(\mathbf{x}) < q0\,; \quad (i = 1, 2, \ldots, n)\,,$$

respecting eventual integrity conditions and optimizing the objective functions:

$$z_j(\mathbf{x})\,, \quad j = 1, 2, \ldots, m\,.$$

The general form of a multiobjective mathematical program is as follows:

$$\begin{cases} \text{Optimize} \quad [z_1(\mathbf{x}), z_2(\mathbf{x}), \ldots, z_m(\mathbf{x})] \\ h_i(\mathbf{x}) < q0 \quad (i = 1, \ldots, n) \\ \mathbf{x} \in X \end{cases}$$

A multiobjective mathematical program is in fact a multicriteria decision problem where [14]: (i) $A = \{\mathbf{x}: h_i(\mathbf{x}) \leq 0, \forall\, \mathrm{i}\} \subset \mathbf{R}^p$ is the set of decision alternatives and (ii) $F = \{z_1(\mathbf{x}), z_2(\mathbf{x}), \cdots, z_m(\mathbf{x})\}$ is a set of criteria where each criterion is expressed by an objective function in terms of the decision variables.

Sensitivity/Robustness Analysis The analysts should examine, through *sensitivity analysis*, the stability of results with respect to the variation of different parameters. Sensitivity analysis is the basis for *robustness analysis*. There are several proposals to enhance GIS-based multicriteria decision making with sensitivity analysis procedures (e. g., [3]). Robustness analysis in multicriteria decision making is a relatively new research topic. Proposals for enhancing GIS-based multicriteria decision making with robustness analysis are still lacking.

Final Recommendation The final recommendation in multicriteria analysis may take different forms according to the manner in which a problem is stated. Roy [12] identifies four types of results corresponding to four ways for stating a problem: (i) *choice*: selecting a restricted set of alternatives, (ii) *sorting*: assigning alternatives to different predefined categories, (iii) *ranking*: classifying alternatives from best to worst with eventually equal positions or (iv) *description*: describing the alternatives and their follow-up results.

Key Applications

GIS-based multicriteria analysis is used in a wide range of decision and management situations. In a recent literature review, Malczewski [10] enumerates about 319 papers devoted to GIS-based multicriteria analysis between 1990 and 2004. The complete list of these papers is available at http://publish.uwo.ca/~jmalczew/gis-mcda.htm.

Environment Planning and Ecology Management

GIS-multicriteria evaluation has been intensively used in environment planning and ecology management. Most analyses within this application area concern land suitability, resource allocation, plan/scenario evaluation, impact assessment and site search/selection problems.

Transportation

Within the transportation application domain, GIS-based multicriteria evaluation is used essentially in vehicle routing and scheduling, and land suitability problems.

Urban and Regional Planning

Major uses of GIS-multicriteria analysis in urban and regional planning concern resource allocation, plan/scenario evaluation, site search/selection and land suitability problems.

Waste Resource Management

The problems tackled in this application domain concern land suitability, plan/scenario evaluation and site search/selection.

Hydrology and Water Resources

In the hydrology and water resources application domain, GIS-multicriteria analysis is used essentially for plan/scenario evaluation. There are also some works for site search/selection and land suitability problems.

Forestry

Major problems tackled within the forestry application domain are land suitability, site search/selection and forestry resources allocation.

Agriculture

The problems considered here are essentially land suitability for different agricultural uses and resources allocation for agricultural activities. Some works are concerned with site search/selection and plan/scenario evaluation problems.

Natural Hazard Management

The problems considered within this application domain mainly concern land suitability and plan/scenario evaluation.

Recreation and Tourism Management

Within this application area, the most treated problem is site search/selection.

Health Care Resource Allocation

Major works in this application domain concern health care site search/selection.

Housing and Real Estate

The problems that are treated here concern land suitability for habitat and real estate, plan/scenario evaluation and site selection for habitation restoration.

Future Directions

There are many important proposals concerning GIS-based multicriteria spatial decision making. However, these proposals present some limitations that prevent them from going beyond the academic contexts. Some of these limitations are cited in the following section.

Integration of Utility-Based Decision Rules

A major part of GIS and multicriteria analysis integration works use utility-based decision rules. However, outranking relation-based decision rules are generally more appropriate to deal with ordinal aspects of spatial decision problems. The natural explanation for this is that the outranking relation-based decision rules have computational limitations with respect to the number of alternatives they consider [11]. One possible solution to facilitate the use of decision rules based outranking relation is to reduce the number of potential alternatives. The idea that is generally used consists of subdividing the study area into a set of homogenous zones which are then used as decision alternatives or as a basis for constructing these alternatives.

Spatial and Temporal Dimensions in Multicriteria Modeling

Two points need to be addressed here: the construction of criteria involving divergent consequences and the modeling of preferences that vary across time and space. In the literature, there are some papers that deal with the construction of criteria based on divergent consequences and the modeling of time-dependent preferences. With respect to GIS-based multicriteria analysis, there are a few papers that take these aspects into account [3].

Fuzzy Spatial Multicriteria Decision Making

Malczewski [10] estimates that 77% of the papers that were published between 1990 and 2004 related to GIS multicriteria analysis used deterministic information. There are several plans to incorporate multicriteria methods supporting imprecision, uncertainty and fuzziness into GIS [6]. The integration of such methods in a geographical information system has the potential to enhance its analytical strength.

Multicriteria Group Spatial Decision Making

Spatial decision problems naturally involve several different kinds of stakeholders. However, the majority of the GIS-multicriteria articles consider individual decision maker's approaches and only a few works (e. g., [5]) are devoted to multicriteria *group spatial decision making*.

Web-Based Multicriteria Spatial Decision Making

There is an increasing interest in the development of Web-based GIS multicriteria evaluation systems [1]. Research on this topic is worthwhile since it promotes the sharing and access of geographical information and facilitates multicriteria collaborative spatial decision making.

Cross References

- ► Data Analysis, Spatial
- ► Decision-Making Effectiveness with GIS
- ► Internet GIS
- ► Multicriteria Spatial Decision Support Systems
- ► Raster Data
- ► Sensitivity Analysis
- ► Temporal GIS and Applications
- ► Uncertainty, Modeling with Spatial and Temporal

► Vector Data
► Web Services, Geospatial

Recommended Reading

1. Carver, S.: Developing web-based GIS/MCE: Improving access to data and spatial decision support tools. In: Thill, J.C.(eds.) Multi-criteria decision-making and analysis: A geographic information sciences approach, pp. 49–75. Ashgate Publishing Ltd., Aldershot, United Kingdom (1999)
2. Chakhar, S., Martel, J.M.: Enhancing geographical information systems capabilities with multi-criteria evaluation functions. J Geogr Inf Decis Anal **7**(2), 47–71 (2003)
3. Feick, R.D., Hall, B.G.: A method for examining the spatial dimension of multicriteria weight sensitivity. Int J Geogr Inf Sci **20**(7), 703–726 (2004)
4. Figueira, J., Greco, S., Ehrgott, M.: Multiple Criteria Decision Analysis: State of the Art Surveys. Springer-Verlag, New York (2005)
5. Jankowski, P., Andrienko, N., Andrienko, G.: Map-centered exploratory approach to multiple criteria spatial decision making. Int J Geogr Inf Sci **15**, 101–127 (2001)
6. Jiang, H., Esatman, J.R.: Applications of fuzzy measures in multi-criteria evaluation in GIS. Int J Geogr Inf Sci **14**(2), 173–184 (2000)
7. Keeney, R.L., Raffia, H.: Decisions with multiple objectives: Preferences and value trade-offs. John Wiley & Sons, New York (1976)
8. Keeney, R.L.: Valued-focused thinking: A path to creative decision. Harvard University Press, Cambridge, USA (1992)
9. Malczewski, J.: GIS and multicriteria decision analysis. John Wiley & Sons, New York (1999)
10. Malczewski, J.: A GIS-based multicriteria decision analysis: A survey of the literature. Int J Geogr Inf Sci **20**(7), 703–726 (2006)
11. Marinoni, O.: A discussion on the computational limitations of outranking methods for land-use suitability assessment. Int J Geogr Inf Sci **20**(1), 69–87 (2006)
12. Roy, B.: Multicriteria methodology for decision aiding. Kluwer Academic Publishers, Dordrecht (1996)
13. Saaty, T.L.: The analytic hierarchy process. McGraw-Hill, New York (1980)
14. Vincke, P.H.: Multicriteria decision-aid. John Wiley & Sons, Chichester (1992)

Multicriteria Spatial Decision Support Systems

SALEM CHAKHAR, VINCENT MOUSSEAU
LAMSADE, University of Paris Dauphine, Paris, France

Synonyms

Spatial multicriteria decision support systems

Definition

A *spatial decision support system* (SDSS) is an interactive, computer-based system designed to support a user or a group of users in achieving a higher effectiveness of decision making while solving a semi-structured spatial decision problem [10]. It lies at the intersection of two major trends in the spatial sciences: *geographic information sciences* and *spatial analysis* [10]. What really differentiates a SDSS and a traditional *decision support system* (DSS) is the particular nature of the geographic data considered in different spatial problems and the high level of complexity of these problems. An effective SDSS requires enhancing conventional DSS with a range of specific techniques and functionalities used especially to manage spatial data. According to [5], a SDSS should (i) provide mechanisms for the input of spatial data, (ii) allow representation of spatial relations and structures, (iii) include the analytical techniques of spatial analysis, and (iv) provide output in a variety of spatial forms, including maps. *Multicriteria spatial decision support systems* (MC-SDSS) can be viewed as part of the broader fields of SDSS. The specificity of MC-SDSS is that it supports *spatial multicriteria decision making*. Spatial multicriteria decision making refers to the use of *multicriteria analysis* (MCA) in the context of spatial decision problems. MCA [7] is a family of operations research tools that have experienced very successful applications in different domains since the 1960s. It has been coupled with geographical information systems (GIS) since the early 1990s for enhanced decision making.

M

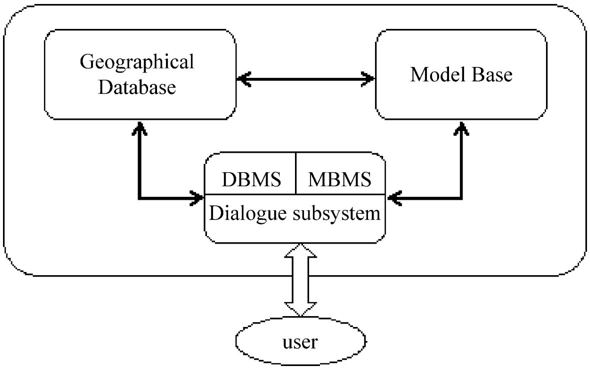
Historical Background

The concept of SDSS has evolved in parallel with DSSs [12]. The first MC-SDSS were developed during the late 1980s and early 1990s [10]. Early research on MC-SDSS is especially devoted to the physical integration of the GIS and MCA. These first tools emphasize interactively and flexibility since GIS and MCA softwares are coupled indirectly through an intermediate system. Later research concerns the development of MC-SDSS supporting collaborative and participative multicriteria spatial decision making [9]. Web-based MC-SDSS is an active research topic which will be the subject of considerable interest in the future [2].

Scientific Fundamentals

General Structure of SDSS/MC-SDSS

A typical SDSS contains three generic components [10] (see Fig. 1): a database management system and geographical database, a model-based management system and model base, and a dialogue generation system. The data management subsystem performs all data-related tasks; that is, it stores, maintains, and retrieves data from the



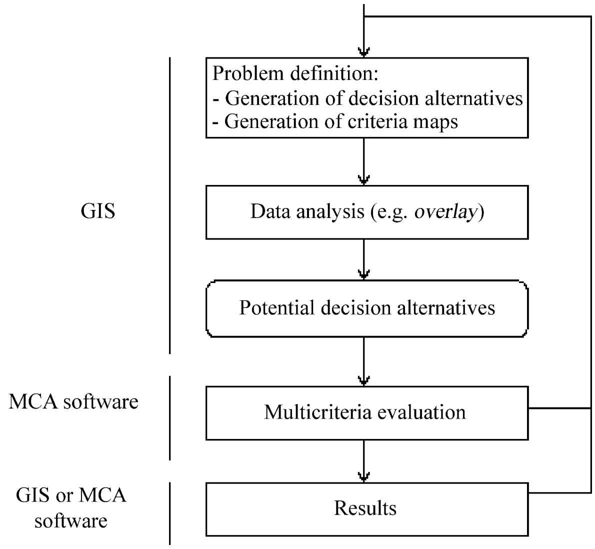
Multicriteria Spatial Decision Support Systems, Figure 1 General structure of SDSS [10]

database, extracts data from various sources, and so on. It provides access to data as well as all of the control programs necessary to get those data in the form appropriate for a particular decision making problem. The model subsystem contains the library of models and routines to maintain them. It keeps track of all possible models that might be run during the analysis as well as controls for running the models. The model base management system component provides links between different models so that the output of one model can be the input into another model. The dialogue subsystem contains mechanisms whereby data and information are input into the system and output from the system. These three components constitute the software portion of the SDSS. A fourth important component of any decision support system is the user which may consist of simple users, technical specialists, decision makers and so on.

MC-SDSS can be viewed as a part of a broader field of SDSS. Accordingly, the general structure of a MC-SDSS is the same as that of a SDSS. However, the model-based management system is enhanced to support multicriteria spatial modeling and the model base is enriched with different multicriteria analysis techniques.

GIS and Multicriteria Analysis Integration Modes

The conceptual idea on which most of GIS-based multicriteria analysis relies is to use the GIS capabilities to prepare an adequate platform for using multicriteria methods [3] (see Fig. 2). The GIS-based multicriteria analysis starts with the problem identification, where the capabilities of the GIS are used to define the set of feasible alternatives and the set of criteria. Then, the *overlay* procedures are used in order to reduce an initially rich set of alternatives into a small number of alternatives which are easily evaluated by using a multicriteria method. Finally, the drawing and presenting capabilities of the GIS are used to present results.

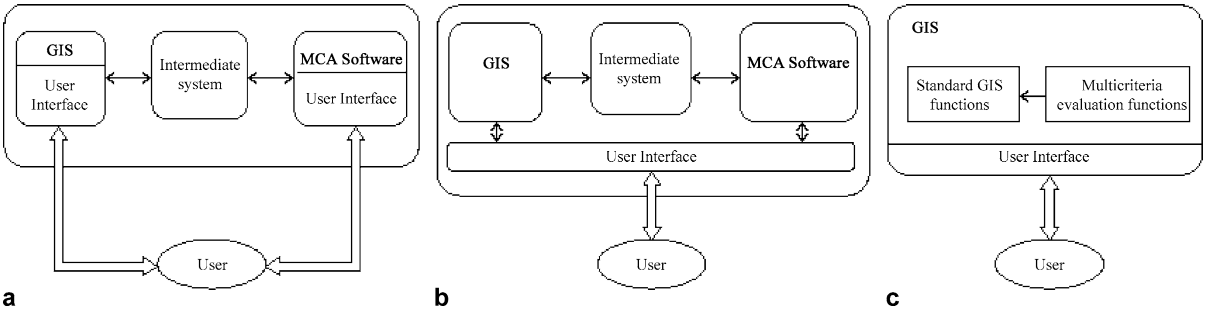


Multicriteria Spatial Decision Support Systems, Figure 2 Conceptual schema for GIS and multicriteria analysis integration

Physically, there are four possible modes to integrate GIS and multicriteria analysis tools [3,10,13]: (i) no integration, (ii) loose integration, (iii) tight integration, and (iv) full integration. The first mode corresponds to the situation that dominated until the late 1980s, when the GIS and multicriteria analysis were used independently to deal with spatial problems. The next three modes correspond to increasing levels of complexity and efficiency (see Fig. 3).

Loose Integration Mode The integration of GIS software and a stand-alone multicriteria analysis software application is made possible by the use of an intermediate system. The intermediate system permits the reformulation and restructuring of the data obtained from the overlapping analysis performed through the GIS, and is converted into a form that is convenient to the multicriteria analysis software. The other parameters required for the analysis are introduced directly via the multicriteria analysis software interface. The results of the analysis—totally made in the multicriteria analysis software—may be visualized by using the presentation capabilities of the multicriteria analysis package, or feedback to the GIS part, via the intermediate system, for display and, eventually, for further manipulation. Each part has its own database and its own interface, which limits the user-friendliness of the system.

Tight Integration Mode In this mode, a particular multicriteria analysis method is directly added to the GIS



Multicriteria Spatial Decision Support Systems, Figure 3 GIS and multicriteria loose (**a**), tight (**b**) and full (**c**) integration modes [3,10]

software. The multicriteria analysis method constitutes an integrated but autonomous part with its own database. The use of the interface of the GIS part alone increases the interactivity of the system. This mode is the first step toward a complete GIS-multicriteria analysis integrated system. Yet, with the autonomy of the multicriteria analysis method, the interactivity remains a problem.

Full Integration Mode The third mode yields itself to a complete GIS-multicriteria analysis integrated system that has a unique interface and a unique database. Here, the multicriteria analysis method is activated directly from the GIS interface, as any GIS basic function. The GIS database is extended so as to support both the geographical and descriptive data, on the one hand, and the parameters required for the multicriteria evaluation techniques, on the other hand. The common graphical interface enhances the user-friendless of the global system.

GIS and Multicriteria Analysis Interaction Directions

It is possible to distinguish five different directions of interaction [11,13]: (i) no interaction, (ii) one-directional interaction with the GIS as the main software (iii) one-direction interaction with the multicriteria tool as the main software, (iv) bi-directional interaction, and (v) dynamic interaction. One-directional interaction provides a mechanism for importing and exporting information via a single flow that originates either in the GIS or multicriteria software. This type of interaction can be based on GIS or multicriteria as the main software. In the bi-directional interaction approach, the flow of data and information can both originate and end in the GIS and multicriteria decision making modules. Dynamic integration allows for a flexible moving of information back and forth between the GIS and multicriteria modules according to the user's needs.

Design of a MC-SDSS

Different frameworks for designing MC-SDSS have been proposed in the literature [3,9,10]. Apart from differences in GIS capabilities and multicriteria techniques, most of these frameworks contain the major components introduced earlier. In the rest of this section, a revised version of the framework proposed in [3] is presented. This framework is conceived of in such a way that it supports GIS-MCA integration and is also open to incorporating any other OR/MS tool into the GIS (see Fig. 4).

Spatial Database Management System The spatial database management system is an extension of the conventional database base management system. It is used specially to manage spatial data.

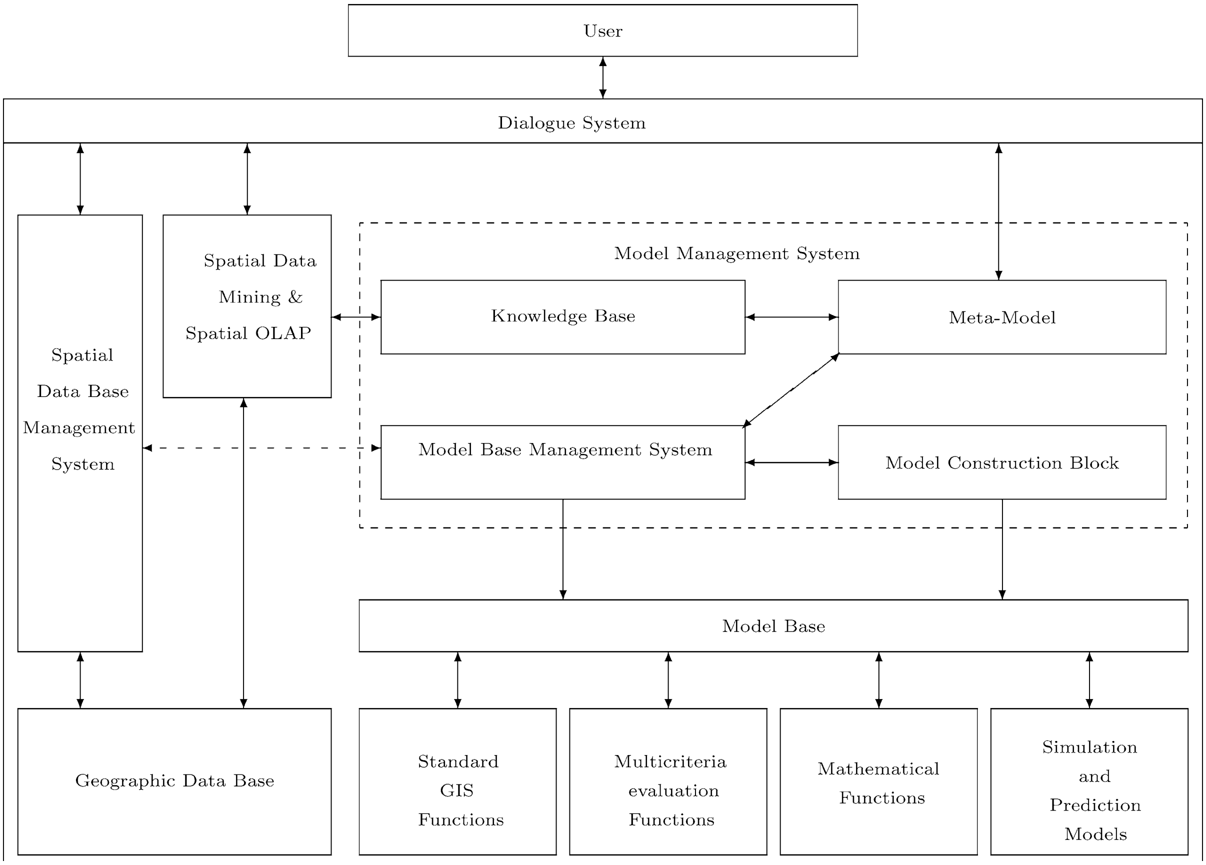
Geographic Database The geographic database is an extended GIS database. It constitutes the repository for both (i) the spatial and descriptive data, and (ii) the parameters required for the different OR/MS tools.

Model Base The model base is the repository of different analytical models and functions. These functions include the basic functions of a GIS, including statistical analysis, overlaying, spatial interaction analysis, network analysis, etc. The model base also contains other OR/MS models and perhaps the most important ones are multicriteria analysis tools. The system is also open to including any other OR/MS tool (e. g., mathematical models, simulation and prediction models, etc.), or any other ad hoc model developed by the model construction block.

Model Management System The role of this component is to manage the different analysis models and functions. The model management system contains four elements: the meta-model, the model base management system, the model construction block and the knowledge base.

Meta-Model This element is normally an expert system used by the decision maker to explore the model base. This exploration enables the decision maker to perform a "what-if" analysis and/or to apply different analytical

M



Multicriteria Spatial Decision Support Systems, Figure 4 A design of a multicriteria SDSS

functions. The meta-model uses a base of rules and a base of facts incorporated into the knowledge base. The notion of the meta-model is of great importance in the sense that it makes the system open for the addition of any OR/MS analysis tool. This requires the addition of the characteristics of the analytical tool to the base of rules, and, of course, the addition of this model to the model base.

Knowledge Base The knowledge base is the repository for different pieces of knowledge used by the meta-model to explore the model base. The knowledge base is divided into a base of facts and a base of rules. The base of facts contains the facts generated from the model base. It also contains other information concerning the uses of different models, the number and the problems to which each model is applied, etc. The base of rules contains different production rules which are obtained from different experts, or automatically derived by the system from past experiences. For instance, this base may contain the following rule: *If the problem under study is the concern of many parties having different objective functions, then the appropriate tool to apply is multicriteria analysis (MCA).*

Model Base Management System The role of the model base management system is to manage, execute and integrate different models that have been previously selected by the decision maker through the use of the meta-model.

Model Construction Block This component gives the user the possibility to develop different ad hoc analysis models for some specific problems. The model that is developed can then be added directly to the model base and its characteristics can be introduced into the knowledge base.

Spatial Data Mining and Spatial on Line Analytical Processing *Data mining* and *on line analytical processing* (OLAP) have been used successfully to extract relevant knowledge from huge traditional databases. Recently, several authors have been interested in the extension of these tools in order to deal with huge and complex spatial databases. In particular, [6] underlines that *spatial data mining* is a very demanding field that refers to the extraction of implicit knowledge and spatial relationships which are not explicitly stored in geographical databases. The same author adds that *spatial OLAP* technology uses multidimensional views of aggregated, prepackaged and structured spatial data to give quick access to information. Incorporating spatial data mining and spatial OLAP into the MC-SDSS will undoubtedly ameliorate the negative impacts when the quality of data is a problem and, consequently, add value to the decision-making process.

Dialogue System The dialogue system represents the interface and tools used to support the dialogue between the user and the MC-SDSS. It permits the decision maker to enter queries and to retrieve the results.

Key Applications

MC-SDSS have been used in a wide range of practical applications of spatial multicriteria decision making problems. They include nuclear waste disposal facility location, solid waste management, land-use planning, corridor location, water resource management, residential site development, health care resource allocation and land suitability analysis. In the rest of this section, a brief description of some SDSS are provided.

- OSDM (Open Spatial Decision Making) [1] is an Internet-based MC-SDSS designed to support the selection of suitable sites for radioactive waste disposal by the public in Great Britain. An important characteristic of OSDM is that it does not require prior knowledge of GIS or MCA.
- Spatial Groupe Choice (SGC) [9] is a GIS-based decision support system for collaborative spatial decision support making. The system has been successfully used for residential site selection in the Duwamish Waterway and surrounding areas, and for health care resource allocation.
- IDRISI/Decision Support is a built-in decision support module for performing multicriteria decision analysis. This system has been applied in different real world applications. The case study described in [10] illustrates the use of the system for analyzing land suitabillity for a housing project in Mexico.
- DOCLOC has been designed for aiding health practitioners in the selection of practices in the state of Idaho [8]. One limitation to this system is the use of the loose coupling strategy.
- Collaborative Planning Support System (CPSS) [14] provides an example of a system employing multiobjective fuzzy decision analysis. It is a multicriteria collaborative spatial decision support system for sustainable water resource management.

Future Directions

Use Full Integration Modes

The first limitation concerning MC-SDSS is relative to the integration mode adopted. In fact, most of the proposed works use loose or tight integration modes. One possible solution to permit full integration is to identify a restricted set of *multicriteria evaluation functions* and incorporate them into the GIS [3]. These functions represent elementary operations required to implement the major part of multicriteria methods. This integration strategy avoids the necessity of programming the different multicriteria methods. In addition, it permits a full integration since the multicriteria evaluation functions are generic and can easily be incorporated in the available commercial GIS.

Incorporation of Large Number of Multicriteria Methods

It is well established that each multicriteria method has its advantages and disadvantages. This means that a given method may be useful in addressing some problems but not in others. One intuitive solution to this problem is to incorporate as many multicriteria methods in the MC-SDSS as possible. However, this idea has several limitations: (i) the obtained system is not flexible enough, (ii) it requires a considerable effort for programming the different methods, and (iii) there is no way to develop "personalized" methods. The integration strategy proposed in the previous paragraph permits the overall system to handle this limitation. In fact, the multicriteria evaluation functions are defined in a generic way and can be used to implement different multicriteria methods or even to create ad hoc methods adapted to the problem under consideration.

Formal Methodology to Select the Multicriteria Method to Apply

Employing a large number of multicriteria methods in the MC-SDSS permits the extension and reinforcement of the analytical potential of the GIS. However, a new problem appears: how to choose the method to use in a given problem? There are generally three possible solutions to the multicriteria method selection problem: (i) the use of a classification tree (ii) the use of a multicriteria method, and (iii) the use of an expert system or a decision support system. It is thought that the last solution is more appropriate from the perspective of GIS and multicriteria analysis integration. The development of a rule-based system needs the designers to work out (i) the characterization of the spatial decision problems, the multicriteria methods and the decision maker(s) (ii) the identification and quantification of knowledge about multicriteria methods, and (iii) the establishment of correspondences among the elements enumerated in (i). The result is a collection of rules. These last ones are then used by the inference system as a basis for selecting the most appropriate method.

Choice of the Standardization/Weighting Techniques

Among the problems that are not sufficiently treated in GIS-based multicriteria systems is the selection of the standardization and the weighting techniques. There are many different standardization/weighting techniques that can be used in MC-SDSS. It is important to note that different standardization/weighting techniques may lead to different results. The development of a formal framework for aiding the decision maker during the selection of the standardization/weighting technique—similar to the one proposed for the selection of the multicriteria method—is a good initiative.

Developing a Multicriteria Spatial Modeling Environment

The use of multicriteria analysis in the GIS is complicated by the lack of an appropriate multicriteria spatial modeling environment. A possible solution is to develop a script-like programming language that supports the different multicriteria evaluation functions. DMA, *decision map algebra*, proposed in [4] and inspired from Tomlin's [15] map algebra, seems to be a good starting point.

Web-Based Multicriteria Spatial Decision Making

Web-based MC-SDSS is a recent and active research topic [2]. This is particularly important since it permits the sharing of geographical information and facilitates multicriteria collaborative spatial decision making.

Cross References

► Data Analysis, Spatial
► Multicriteria Decision Making, Spatial

Recommended Reading

1. Carver, S.J.: Open spatial decision making on the Internet. School of Geography, University of Leeds, Yorkshire, England (1996)
2. Carver, S.J.: Developing web-based GIS/MCE: improving access to data and spatial decision support tools. In: Thill, J.C. (eds.) Multi-criteria decision-making and analysis: A geographic information sciences approach, pp. 49–75. Ashgate, Aldershot, England (1999)
3. Chakhar, S., Martel, J.M.: Enhancing geographical information systems capabilities with multi-criteria evaluation functions. J. Geogr. Inf. Dec. Anal. **7**(2), 47–71 (2003)
4. Chakhar S., Mousseau, V.: DMA: An algebra for multicriteria spatial modeling. In: Proccedings ICA Workshop on Geospatial Analysis and Modeling, 8 July, pp. 155–185, Vienna, Austria (2006)
5. Densham, P.J.: Spatial decision support systems. In: Maguitre, D.J., Goodchild, M.F., Rhind, D. (eds.) Geographical information systems: Principles and applications, vol. 1, pp. 403–412, Longman, London (1991)
6. Faiz, S.: Managing geographic data quality during spatial data mining and spatial olap. data warehousing and data quality. GIM Int **14**(12), 28–31 (2000)
7. Figueira, J., Greco, S., Ehrgott, M.: Multiple criteria decision analysis: State of the art surveys. Springer-Verlag, New York (2005)
8. Jankowski, P., Ewart, G.: Spatial decision support system for health practitioners: Selecting a location for rural health practice. Geographical Systems **3**(2), 297–299 (1996)
9. Jankowski, P., Nyerges, T.L., Smith, A., Moore, T.J., Horvath, E.: Spatial group choice: A SDSS tool for collaborative spatial decision-making. International Journal of Geographical Information Systems **11**, 566–602 (1997)
10. Malczewski, J.: GIS and multicriteria decision analysis. John Wiley & Sons, New York (1999)
11. Malczewski, J.: A GIS-based multicriteria decision analysis: A survey of the literature. International Journal of Geographical Information Science **20**(7), 703–726 (2006)
12. G.M. Marakas, G.M.: Decision support systems in the 21st centry, 2nd edn. Prentice-Hall, Upper Saddle River (2003)
13. Nyerges, T.L.: Coupling GIS and spatial analytic models. In: Breshanan, P., Corwin, E., Cowen, D. (eds.) Proceedings of the 5th International Symposium on Spatial Data Handling. pp. 534–543. Charleston, SC. Humanities and Social Sciences Computing Laboratory, University of South Carolina (1992)
14. Simonovic, S.P., Bender, M.J.: Collaborative planning-support system: An approach for determining evaluation criteria. Journal of Hydrology **177**(3–4), 237–251 (1996)
15. Tomlin, C.D.: Geographic information systems and cartographic modeling. Prentice Hall, Englewood Cliffs, New Jersey, USA (1990)

Multi-Dimensional Access Structures

► Indexing, High Dimensional

Multidimensional Index

► Indexing, Hilbert R-tree, Spatial Indexing, Multimedia Indexing

Multi-Dimensional Indexing

► Indexing, High Dimensional

Multi-Dimensional Mapping

► Space-Filling Curves

Multi-Dimensional Time Series Similarity

► Trajectories, Discovering Similar

Multilateration

► Indoor Localization

Multimedia Atlas Information Systems

LORENZ HURNI
Institute of Cartography, ETH Zurich, Zurich, Switzerland

Synonyms

Cartographic information system; Atlas information system; Atlas, electronic; Atlas, interactive; Atlas, multimedia; Atlas, virtual; Google Earth; Digital Earth

Definition

Multimedia atlas information systems (MAIS) are systematic, targeted collections of spatially related knowledge in electronic form, allowing a user-oriented communication for information and decision-making purposes. As in a conventional atlas, a MAIS mainly consists of a harmonized collection of maps with different topics, scales, and/or from different regions. The maps usually come in standardized scales or degrees of generalization, respectively. The different map types have a common legend and symbolization. The access to the maps is granted through thematic or geographic indexes. MAIS dispose of special interactive functions for geographic and thematic navigation, querying, analysis and visualization in 2D and 3D mode. Unlike in many geographic information systems (GIS) applications, the data in MAIS is cartographically edited and the functionality is intentionally limited in order to provide a user-targeted set of data as well as adapted analysis and visualization functions. In multimedia atlases, additional related multimedia information, like graphics, diagrams, tables, text, images, videos, animations, and audio documents, are linked to the geographic entities. Efficient management of the increasing amount of information led to the development of database-driven MAIS. Most MAIS are based on CD-ROM, DVD or increasingly on web technologies (intranet, internet).

Historical Background

The technological leap, which caused the transition from analog to digital cartography in the 1980s, has also stimulated the development of interactive atlases. GIS, computer aided design (CAD) systems, desktop publishing (DTP) systems and the thereby-evoked releases of geometric and thematic cartographic data were the catalysts of both digital and interactive cartography. It is disputed which atlas was the first digital one: Some authors claim an early version of the *Electronic Atlas of Canada* was the first digital atlas [1], others consider that it was the *Electronic Atlas of Arkansas* [2]. Early digital atlases had a rather limited functionality, like name search, zoom, and layer selection. Other atlases like the PC version of the *National Atlas of Sweden* were based on commercial GIS software. In the following years, interactive atlases were evolving with respect to content, data and technology. In several countries national atlases on CD-ROM were produced, either as a digital version of a conventional paper atlas (such as the *National Atlas of Germany*), or as entirely interactive version (such as the *Atlas of Switzerland*). In the late 1990s, national mapping authorities began to publish their topographical map series on CD-ROM/DVD. A third group of atlases are counterpieces to conventional world or school atlases, such as *Microsoft Encarta*, which, however, is today integrated in the *Encarta* encyclopedia. Technologically, the first atlases were based on raster data maps like most of the electronic national map series. Modern interactive atlases make use of vector data sets and/or statistical data which are symbolized and visualized on the fly (e. g., the *Tirol Atlas*). The atlases evolved from CD-ROM, then DVD to web-based or combined interactive atlases.

Multimedia Atlas Information Systems, Table 1 Aspects of cartographic expression forms, after [3]

Aspects of cartographic expression forms	Ordering of aspects
Display media	Print, screen, projection
Dimension of representation	2D, pseudo-3D, 3D
Degree of dynamics	Static, cinematographic, dynamic
Degree of interaction	Noninteractive, partially interactive, interactive
Channels of representation	Visual, acoustic, haptic
User–map relation	Separating, integrative, amplification of reality

Scientific Fundamentals

For the case of interactive maps on new media, the classical graphical variables and their expressions are extended as shown in Table 1 [3].

The added values and advantages of MAIS compared to paper atlases can be summed up as follows: interactivity, navigation, maps as interface, exploration, cus-

Multimedia Atlas Information Systems, Table 2 Main functions in a multimedia atlas information system (MAIS) [6]

Function Groups	Function subgroups	Functions
General functions		Mode selection, language selection, file import/export, printing, placing bookmarks, hot spots, forward/backward, settings (preferences), tooltips, display of system state, help, imprint, home, exit
Navigation functions	Spatial navigation	Spatial unit selection, enlarge/reduce of map extend (zoom in, zoom out, magnifier), move map (pan, scroll), reference map/globe, map rotation, determination of location (coordinates, altitude), line of sight and angle, placement of pins, spatial/geographical index, spatial/geographical search, tracking
	Thematic navigation	Theme selection and change, index of themes, search by theme, theme favorites
	Temporal navigation	Time selection (positioning of time line, selection of time period), animation (start/stop etc.)
Didactic functions	Explanatory functions	Guided tours, preview, explanatory texts, graphics, images, sounds, films
	Self-control functions	Quizzes, games
Cartographic and visualization functions	Map manipulation	Switch on/off layers, switch on/off legend categories, modification of symbolization, change of projection
	Redlining	Addition of user defined map elements, addition of labels (labeling)
	Explorative data analysis	Modification of classification, modification of appearance/state (brightness, position of sun), map comparison, selection of data
GIS functions	Space and object oriented query functions	Spatial query/position query (coordinates query/query of altitude), measurement/query of distance and area, creating profile
	Thematic query functions	Thematic queries (data/attribute queries), access to statistical table data
	Analysis functions	Buffering, intersection, aggregation and overlapping (transparent overlapping/fading), terrain analysis (exposition, slope etc.)

tomized/customizable to user's need, updatable, dynamics/animation, multimedia integration [4,5].

The degree of interactivity, a very significant element of the usability of a cartographic application, is mainly based on the richness of available cartographic functions. Table 2 shows the most important functions, arranged in five main groups [6].

Complementary, MAIS can be characterized according to the basic concepts as shown in Table 3. Today, most atlases still consist of raster and vector base data, but a transition to relational or object-oriented vector data can be observed. Most atlases are still bound to classic computer interfaces like keyboards, mice and screens. Internet and mobile technologies will increase the degree of system distributivity. With respect to interactivity, atlases are arranged into three groups: View-only atlases, interactive atlases and analytical atlases [7]. The latter can be subdivided into simple, constructive, and automatic analytical atlas types [8]. Furthermore, many atlases serve no longer as a main, but as one out of several possible interfaces to the data, e. g., in the Encarta encyclopedia.

Today's MAIS comprise of basic topographic and thematic data and software allowing the creation of maps on demand, as in GIS [9]. However the differences between MAIS and GIS can be perceived when comparing three approaches for applying GIS to the development of MAIS [10,11]. The concept "multimedia in GIS" proposes the integration of multimedia functionality in GIS, mainly at the cost of user-friendliness. "GIS in multimedia" incorporated explicitly defined and developed GIS functions in a cartographic multimedia environment. The third concept "GIS analysis for multimedia atlases" combines a GIS, the authoring system and a multimedia map extension (GIS data converter) in one common multimedia atlas development environment. Table 4 shows the main differences between GIS and MAIS [12].

Key Applications

World Atlases, School Atlases

Interactive world atlases mainly consist of physical (and some thematic) maps of the world with search and index functions. The most prominent example is Microsoft's *Encarta* atlas which is now integrated in the interactive *Encarta* encyclopedia. This allows the linking of places with multimedia elements of the encyclopedia and vice versa. A special version of the world atlases are school atlases which also include more thematic maps and numerous exemplary maps for didactic purposes. An example is the Austrian atlas *Geothek* by Ed Hölzel publishers.

Multimedia Atlas Information Systems, Table 3 Main characteristics and concepts of MAIS [8]. *OO* object-oriented, *PDA* Portable Digital Assistant, *UMN* University of Minnesota, *WMS* Web Map Service, *LBS* location-based services

Main characteristic	Characteristic /functionality	Subgroups/remarks	Examples
Data type and modeling	Raster		Raster GIS, map layers in raster format
	Vector	Sequentially attributive (file-oriented)	DTP files with attributes
		Relational-topological	Database-based system (geometry and thematic data)
		Object-oriented-topological	OO-geo-databases
Medium, communication channel	Text		Keyboard, alphanumeric output
	Language		Voice output in car navigation systems
	Screen	Stationary screens	Computer screen
		Portable screens	PDA, mobile phone
Degree of system distributivity	Off-line	Local system (client based)	MAIS on CD/DVD
	On-line (1 : 1)	Client/server based	UMN Map Server
	Distributed (1 : n)	One client/several server	WMS
	Multiple distributed (n : m)	Several clients/several server	LBS networks
Degree of interactivity	View only	Display of prepared maps	Information maps on the Internet
	Interactive	Queries by criteria, adjustment of output/display	MAIS like *Atlas of Switzerland* V1
	Simple analytical	Combined queries, more complex (GIS-like) analysis functions	MAIS with GIS functions, data however prepared (*Atlas of Switzerland* V2 partially)
	Constructive analytical	Direct processing of user data, design possibilities	Web-GIS, projection web services
	Automatic analytical	Automatic data analysis and rule-based processing	Cartographic real-time web information systems, e. g., on-line avalanche maps, radar precipitation maps, egocentric real-time information display on LBS, online generalization
Priority of cartographic functionality	Map information systems	Map functions as main interaction tools	MAIS, web map information systems
	General information systems	Map functions as further query and export/display possibility	Digital encyclopedias (e. g., Encarta), environmental information systems, Real estate portals

M

National Atlases, Regional Atlases

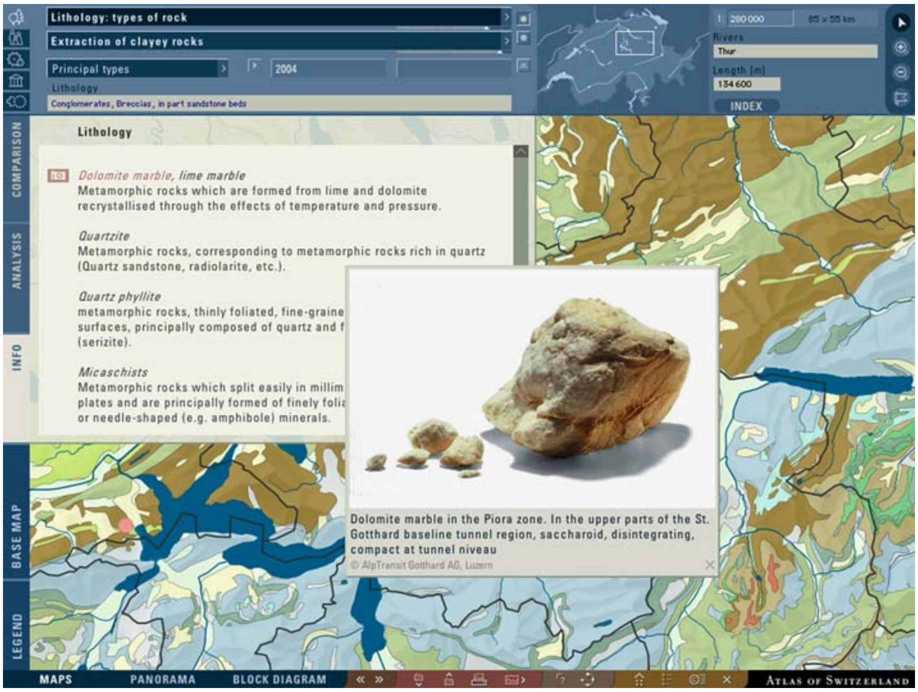
National and regional atlases depict a country or a region in a broad variety of mainly thematic maps. Today many national atlases have been converted from the printed to the interactive form. There also exist mixed versions like the *National Atlas of Germany* which consists of a series of theme books and accompanying CD-ROMs with the full text, the maps plus some additional interactive maps [13]. An example for an entirely digital atlas is the DVD-based *Atlas of Switzerland* which consists of 1,000 interactive maps derived on the fly from digital topographic, environmental and statistical base data, combined with multimedia elements (Fig. 1) [14].

Topographic Atlases

Many national or state mapping authorities publish their topographic map series on CD-ROM or DVD. In mostcases, the maps are stored in raster format, but enriched with place names and vector line data for routes and trails. Many products offer the possibility of importing own data like GPS tracks or drawing map overlays. Simple analyses like measurement functions, profiles and 3D displays are possible. Examples are e. g., the *TOPO! Interactive Maps* with US Geological Survey map sheets, published by National Geographic and the *Swiss Map* DVDs with the *Swiss National Map Series*. Other atlases display georeferenced satellite or aerial images.

Thematic Atlases, Statistical Atlases

Numerous atlases cover specific thematic topics like geology, hydrology, climate, planning, history, etc., both in 2D and 3D (Fig. 2). Statistical atlases allow the visualization of statistical data as choropleth or diagram maps, usually



Multimedia Atlas Information Systems, Figure 1 Example of a multimedia national atlas: lithology map in the *Atlas of Switzerland*, combined with text and image information. © Atlas of Switzerland

Multimedia Atlas Information Systems, Table 4 Differences between GIS and multimedia atlas information systems (*MAIS*); adapted after [12]

	GIS	MAIS
Target users	Experts	Nonexperts (and experts)
Use of interface	Complex	Easy
Control of functions and data	By users	By authors
Guidance	Minimal	Distinct
Flow of information	Unstructured	Structured (narrative)
Main focus	Handling, analysis and presentation of data	Visualization of themes
Data	Raw, not integrated	Edited, integrated
Data model	Primary model	Secondary model
Covered area	Open	Usually predefined: regional, national
Computation time	Short to long	Short
Purpose	Open for any kind of data and analysis	Specific purpose

on the basis of administrative boundaries (e. g., the *Geoclip* statistical atlas web service).

Future Directions

A major focus will be the further development of geodata models and structures. Up to now, geodata have been managed and processed in relatively specialized systems like GIS. Data are enriched with graphical attributes for cartographic visualization and thematic attributes. In the future this attributation will be handled the other way round: Thematic data will be stored in standardized, distributed databases and they will additionally be annotated with spatial information, i. e., they will be georeferenced. Search engines could be equipped with a geographical search function.

Specific cartographic functions will be developed further, e. g., automatic generalization functions, rule-based display functions or analysis functions. Real-time data, for instance, will be analyzed automatically and visualized on the fly. The integration of user-generated data will be simplified and a MAIS will become a collaborative platform that can constantly be maintained and updated by the users.

Cross References

- ► Data Analysis, Spatial
- ► Distributed Geospatial Computing (DGC)
- ► Exploratory Visualization
- ► Scalable Vector Graphics (SVG)
- ► Visualizing Constraint Data
- ► Web Feature Service (WFS) and Web Map Service (WMS)

Recommended Reading

1. Ormeling, F.: Atlas information systems. In: Proceedings of the 17th International Cartographic Conference, Barcelona, 3–9 September 1995, pp. 2127–2133

Multimedia Atlas Information Systems, Figure 2 Example of a 3D atlas: 3D display of evaporation data as overlay on a digital elevation model, combined with a satellite image and atmospheric effects. The user-defined profile shows both a topographic and thematic section (*lower right*). © Atlas of Switzerland

M

2. Siekierska, E., Williams, D.: National atlas of Canada on the internet and schoolnet. In: Köbben, B., Ormeling, F., Trainor, T. (eds.) Seminar on Electronic Atlases II, pp. 19–23. ICA Commission on National and Regional Atlases, Prague (1996)
3. Buziek, G.: Eine Konzeption der kartographischen Visualisierung. Habilitation thesis, University of Hannover (2002)
4. Ormeling, F.: Functionality of electronic school atlases. In: Köbben, B., Ormeling, F., Trainor, T. (eds.) Seminar on Electronic Atlases II, pp. 33–39. ICA Commission on National and Regional Atlases, Prague (1997)
5. Borchert, A.: Multimedia atlas concepts. In: Cartwright, W., Peterson, M., Gartner, G. (eds.) Multimedia Cartography, pp. 75–86. Springer, Berlin Heidelberg New York (1999)
6. Cron, J.: Graphische Benutzeroberflächen interaktiver Atlanten – Konzept zur Strukturierung und praktischen Umsetzung der Funktionalität; mit Kriterienkatalog. Diploma thesis, Hochschule für Technik und Wirtschaft, Dresden and Institute o Cartography, ETH Zurich (2006). Available via http://www.ika.ethz.ch/teaching/Diplomarbeit-Cron.pdf; http://www.ika.ethz.ch/teaching/Diplomarbeit-Cron-Katalog.pdf
7. Siekierska, E.M., Taylor, D.R.F.: Electronic mapping and electronic atlases: new cartographic products for the information era—the electronic atlas of Canada. CISM J. ACSGC **45**, 11–21 (1991)
8. Hurni, L.: Interaktive Karteninformationssysteme – quo vaditis? Kartogr. Nachr. **56**, 136–142 (2006)
9. Da Silva Ramos, C., Cartwright, W.,: Atlases from paper to digital medium. In: Stefanakis, E., Peterson, M.P., Armenakis, C., Delis, V.(eds.) Geographic Hypermedia–Concepts and Systems. Lecture Notes in Geoinformation and Cartography:, pp. 97–19. Springer, Berlin Heidelberg New York (2006)
10. Bär, H.R., Sieber, R.: Towards high standard interactive atlases: the GIS and multimedia cartography approach. In: Proceedings of the 19th International Cartographic Conference, Ottawa, 14–21 August 1999, pp. 235–241
11. Schneider, B.: GIS Functionality in Multimedia Atlases: Spatial Analysis for Everyone. In: Proceedings of the 20th International Cartographic Conference Beijing, 6–10 August 2001, pp. 829–840
12. Schneider, B.: Integration of Analytical GIS-functions in Multimedia Atlas Information Systems. In: Proceedings of the 19th International Cartographic Congress Ottawa, 14–21 August 1999, pp. 243–250
13. Hanewinkel, C., Tzschaschel, S.: Germany in maps – a multipurpose toolbox In: Proceedings of the 22th International Cartographic Conference A Coruña, 9–16 July 2005: CD-ROM
14. Sieber, R., Huber, S.: Atlas of Switzerland 2 – A highly interactive thematic national atlas. In: Cartwright, W., Peterson, M.P., Gartner, G.(eds.) Multimedia Cartography. 2nd edn., pp. 161–182. Springer, Berlin Heidelberg New York (2007)

Multimedia Indexing

► Indexing, Hilbert R-tree, Spatial Indexing, Multimedia Indexing

Multiple Resolution Database

► Abstraction of GeoDatabases

Multiple Target Tracking

BAIK HOH, MARCO GRUTESER
Electrical and Computer Engineering/WINLAB,
Rutgers University, Piscataway, NJ, USA

Definition

The problem of associating a set of anonymous position observations with sets of prior observations to construct the traces of several moving objects or targets. More precisely, given n_t position observations at time t and previous observations $n_{t-1}, n_{t-2}, \ldots, n_{t-m}$, construct a set of traces, where each trace only contains the position observations from a single moving object. In a variation of the problem the number of actual moving objects may be greater than each n since objects may disappear and emerge during the observation interval.

Main Text

Multiple hypothesis tracking is one representative algorithm proposed by Reid [2] in 1979 to solve this problem. This algorithm uses a linear Kalman model to represent the movement behavior of each object and to filter observation noise. The algorithm operates in three steps. First it predicts a new system state (which includes predicted positions of each object based on the prior trajectory). Then it generates hypotheses for the assignment of new samples to targets and selects the most likely hypotheses. Finally it adjusts the system state based on the Kalman equations with information from the new samples. In this process, one hypothesis is generated for each permutation of the sample set; each permutation represents one possible association of new observations to the prior targets. The algorithm then selects the hypothesis that minimizes the error between the predicted positions and actual positions across all objects.

The performance of MHT depends on the density of targets, the frequency of observations, the predictability of target movements, and the accuracy of the target movement model. Tracking performance increases with scenarios with low user density or when targets are moving on relatively straight trajectories such as cars driving on a highway. Further tracking performance improves when observations are obtained with higher frequency.

Many variations of this algorithm have been proposed to accommodate target maneuvering, insertion of new targets, and abrupt disappearance of existing targets. If objects move on well-known roadways, multi-hypothesis tracking can be improved by also considering road maps during the linking process [1].

This problem frequently arises when several simultaneously moving objects are tracked by radar. Multi-target tracking algorithms are also a useful tool to understand the privacy of anonymized positions samples in GIS applications. This approach can be viewed as an inference attack; thus, privacy is compromised if these algorithms can recover longer traces from individual samples.

Cross References

► Privacy Preservation of GPS Traces

Recommended Reading

1. Civilis, A., Pakalnis, S.: Techniques for efficient road-network-based tracking of moving objects. IEEE Trans. Knowl. Data Eng. **17**(5), 698–712 (2005) Senior Member-Christian S. Jensen
2. Reid, D.: An algorithm for tracking multiple targets. IEEE Trans. Autom. Control **24**(6), 843–854 (1979)

Multiple Worlds

► Smallworld Software Suite

Multiple-Image Bundle Block

► Photogrammetric Methods

Multirepresentation

► Modeling and Multiple Perceptions

Multi-Resolution Aggregate Tree

IOSIF LAZARIDIS, SHARAD MEHROTRA
Department of Computer Science, University of California, Irvine, CA, USA

Synonyms

MRA-Tree; Ra*-Tree; aR-Tree

Definition

A Multi-resolution Aggregate tree (MRA-tree) is a multi-dimensional indexing structure whose nodes are augmented with aggregate information about the indexed subsets of data. Typically, such indices subdivide space or group data objects; nodes contain routing information for their children nodes, e. g., in the form of spatial partitions (as in quad-trees) or bounding rectangles (as in R-trees). MRA-trees store, in addition to this information, aggregate properties of the indexed entities, e. g., the SUM of their values, the MIN value, etc. Several such aggregates can be stored or alternatively, only those that are often queried.

Main Text

MRA-trees are useful in answering aggregate queries approximately and in a progressive manner. Traditional multi-dimensional indexes help aggregate query answering by quickly gathering all relevant tuples. However, they have the limitation that each of those tuples must be handled individually. Moreover, approximate answers and answer quality guarantees cannot be easily computed. MRA-trees avoid visiting entire subsets of the data since they are summarized adequately at high-level tree index nodes and they can provide deterministic answer quality guarantees since the aggregate characteristics of the data at various levels of resolution are available throughout the tree. By exploring the tree progressively, the answer quality can improve all the way to the exact answer. They can therefore be used when the user specifies either a time deadline or answer quality requirement, trying to optimize the quality and running time, respectively, under these constraints.

Cross References

► Aggregate Queries, Progressive Approximate
► Progressive Approximate Aggregation

Multiscale Databases

► Modeling and Multiple Perceptions

Multi-Type Nearest Neighbor Query

► Trip Planning Queries in Road Network Databases

Mutation

► Geographic Dynamics, Visualization And Modeling

MX-Quadtree

► Quadtree and Octree

National Map Accuracy Standard

► Photogrammetric Products

National Spatial Data Infrastructure (NSDI)

JULIE BINDER MAITRA
Federal Geographic Data Committee (FGDC),
Reston, VA, USA

Synonyms

NSDI

Definition

The NSDI is defined as the "technology, policies, standards, and human resources necessary to acquire, process, store, distribute, and improve utilization of geospatial data" in Office of Management and the Budget (OMB) Circular A-16 (revised 2002), Coordination of Geographic Information, and Related Spatial Data Activities http://www.whitehouse.gov/omb/circulars/a016/a016_rev.html (accessed December 4, 2006).

Historical Background

This historical background recounts the efforts by the FGDC to develop the National Spatial Data Infrastructure (NSDI) in the United States. However, the concepts of the NSDI are scaleable to the local, regional, and global levels and may be applied to other countries.

Two documents set policies for development of the NSDI: (1) Executive Order 12906: Coordinating Geographic Data Acquisition and Access: The National Spatial Data Infrastructure and (2) OMB Circular A-16.

Executive Order 12906, signed by President Bill Clinton on April 11, 1994, launched the initiative to create the NSDI. President George W. Bush amended Executive Order 12906 by issuance of Executive Order 13286 on March 5, 2003. Executive Order 12906/13286 defines NSDI and calls for the development of the NSDI through creation of the National Geospatial Data Clearinghouse, development of geospatial data standards, implementation of a National Digital Geospatial Data Framework, and partnerships for data acquisition with State, local, and tribal governments, the private sector, and other nonfederal organizations.

OMB Circular A-16 (revised 2002) affirms the National Spatial Data Infrastructure (NSDI) as the "technology, policies, standards, human resources, and related activities necessary to acquire, process, distribute, use, maintain, and preserve spatial data." It provides direction to Federal agencies that produce, maintain, or use spatial data, either directly or indirectly in the fulfillment of their mission. It describes the effective and economical use and management of geospatial data assets for the benefit of the government and the nation and management and reporting requirements in the acquisition, maintenance, distribution, use, and preservation of spatial data by the Federal Government.

Geospatial One-Stop, which is one of 24 E-Government initiatives sponsored by OMB to enhance government efficiency and to improve citizen services, was established to accelerate the development of the NSDI. Through Geospatial One Stop, a web portal (geodata.gov) was launched to serve as a single point of access to geospatial data and services. The Geospatial One Stop portal enables data sharing in order to maximize geospatial investments to leverage resources and reduce redundancies.

The Geospatial Line of Business (Geospatial LoB) Presidential Initiative builds on Geospatial One-Stop by identifying opportunities for optimizing and consolidating Federal geospatial-related investments in order to reduce the cost of government and improve services to citizens.

Scientific Fundamentals

Core components of the NSDI are partnerships, geospatial data, metadata, framework, standards and interoperability, and the Clearinghouse/Web portal.

Partnerships

Partnerships between Federal agencies and State, Tribal, and local governments, academic institutions; the private sector, and other communities are necessary to build the NSDI. These communities often hold data that is more current and detailed than data held by the Federal governments. Partnerships with these communities promote meeting the needs of end-users for data and services.

Geospatial Data

Geospatial data are data tied to a position on the Earth through coordinates such as latitude and longitude or through a geographic identifier such as a place name.

Geospatial data are typically grouped by data themes, a legacy of the analog mapping paradigm where different themes (culture, water, elevation contours, etc.) are on different map separates so that they are printed in different colors. However, a single digital file might contain data from many different themes.

Federal agencies with lead responsibilities for NSDI data themes are listed in Appendix E National Spatial Data Infrastructure (NSDI) Data Themes, Definitions, and Lead Agencies, of OMB Circular A-16.

Metadata

Metadata is "data about data" and provides documentation on geospatial data. The minimum requirements for reporting metadata are basic identification information; however, metadata might also cover data quality, data representation (for example, raster or vector), spatial referencing, and entity and attribute information.

Collecting metadata should be integral to any data collection project. As data and information are organizational assets, metadata preserves the value of those assets. Metadata also preserves institutional knowledge that might be lost as people leave an organization. Metadata enables users to evaluate geospatial data for fitness of use. When published to a data clearinghouse, metadata enables search and discovery of geospatial data.

Other geospatial resources may be described in metadata, including online services, data download locations, interactive web applications, documents, and other web-accessible resources.

Framework

Framework data themes refer to seven themes of data identified as critical by many geospatial applications:

1. Cadastral data – Cadastral data describe the geographic extent of past, current, and future right, title, and interest in real property, including above, surface, and below ground and water, and the conceptual structure to support the description of that geographic extent.
2. Digital orthoimagery – Digital orthoimages are georeferenced images of the Earth's surface for which image object displacement caused by sensor orientation, sensor distortions, and terrain relief has been removed. Digital orthoimages have the geometric characteristics of a map and image qualities of a photograph.
3. Elevation – Elevation data may be modeled in various forms, such as in an evenly spaced grid (for example, digital elevation models or DEMs) or as irregularly spaced points (triangulated irregular network, hypsography, or mass points).
   Terrestrial (land) elevation data contain georeferenced digital representations of terrestrial surfaces, natural or manmade, which describe vertical positions above or below a datum. Bathymetric data comprise depths below sea level. Both terrestrial elevation and bathymetric data support the elevation theme of Framework data.
4. Geodetic control – Geodetic control provides a common, consistent, and accurate reference system for establishing coordinates for all geographic data. NSDI framework data may use geodetic control to accurately register spatial data. In the United States, fundamental geodetic control is provided through the National Spatial Reference System (NSRS) managed by the National Oceanic and Atmospheric Administration (NOAA).
5. Government Unit Boundaries – boundary data for government units and legal entities.
6. Hydrography – This data theme includes surface water features such as lakes, ponds, streams or rivers, canals, oceans, and shorelines.
7. Transportation – Transportation data model the geographic locations, interconnectedness, and characteristics of the transportation system. The transportation system includes both physical and non-physical components representing all modes of travel (land, sea, and air) that allow the movement of freight and people between locations.

This list is not intended to undermine the importance of other data themes, such as land cover and geographic names.

The seven data themes provide basic data that can be used in applications, a base to which users can add or attach geographic details and attributes, a reference source for accurately registering and compiling participants' own data sets, and a reference map for displaying the locations and the results of an analysis of other data.

The framework concept also includes procedures, guidelines, and technology to enable participants to build, integrate, maintain, distribute, and use framework data.

These procedures, guidelines, and technology ensure that

- users can depend on accurate, detailed data that can be certified and integrated into the framework to create a trustworthy data source;
- users can update their data holdings from the framework data; and
- users can attach additional information to the framework.

Standards and Interoperability

Standards are common and repeated rules, conditions, guidelines or characteristics for data, and related processes, technology and organization. The NSDI is made possible by the universal use of standards and protocols for data development, documentation, exchange, and geospatial services. OMB Circular A-119 on Federal Participation in the Development and Use of Voluntary Consensus Standards and in Conformity Assessment Activities directs Federal agencies to use voluntary consensus standards whenever possible and participate in voluntary consensus standards bodies.

Key Voluntary Consensus Standards Bodies OMB Circular A-119 defines voluntary consensus standards bodies as "domestic or international organizations which plan, develop, establish, or coordinate voluntary consensus standards using agreed-upon procedures" characterized by openness, balance of interest, due process, an appeals process, and consensus. Key voluntary consensus standards bodies include the ISO, the International Organization for Standardization, American National Standards Institute (ANSI), and the Open Geospatial Consortium (OGC).

ISO, International Organization for Standardization ISO, the International Organization for Standardization, describes itself as a "network of the national standards institutes of 147 countries, on the basis of one member per country." Standards development is carried out by technical committees, including ISO Technical Committee 211, Geographic information/Geomatics, which develops the ISO 19100 series of geographic information standards. Topics covered by the ISO 19100 series of geographic information standards include spatial and topological representation of geographic features, spatial referencing by coordinates and geographic identifiers, feature cataloguing methodology, and metadata.

American National Standards Institute (ANSI) The American National Standards Institute (ANSI) is the U.S. member body of ISO. ANSI is distinct from many other national standards bodies in that it is a non-governmental organization. ANSI facilitates the development of American National Standards (ANS) by accrediting the procedures of standards developing organizations as meeting ANSI requirements for openness, balance, consensus and due process.

ANSI has accredited the procedures of the InterNational Committee for Information Technology Standards (INCITS) to develop standards for Information and Communications Technologies (ICT). INCITS is comprised of many technical committees that develop standards in specific topic areas of ICT, including INCITS Technical Committee L1, Geographic Information Standards (INCITS L1). It is through INCITS L1 that the geospatial community in the U.S. participates in ANSI and ISO standardization activities. INCITS L1 serves as the U.S. Technical Advisory Group (TAG) to ISO Technical Committee 211.

Open Geospatial Consortium (OGC) The Open Geospatial Consortium (OGC) describes itself as follows:

> … an international industry consortium of **337** companies, government agencies and universities participating in a consensus process to develop publicly available interface specifications. OpenGIS® Specifications support interoperable solutions that "geo-enable" the Web, wireless and location-based services, and mainstream IT (from About OGC | OGC®, http://www.opengeospatial.org/ogc, accessed December 4, 2006.

Key OpenGIS® specifications include Web Mapping Service (WMS), Web Coverage Service (WCS), Geography Markup Language (GML), and Web Feature Service (WFS).

The OpenGIS® Web Map Service (WMS) Implementation Specification provides three operations (GetCapabilities, GetMap, and GetFeatureInfo) for the creation and display of registered and superimposed map-like (image) views of information from multiple online sources.

The OpenGIS® Web Coverage Service (WCS) Implementation Specification allows access to geospatial "coverages" (raster data sets) that represent values or properties of geographic locations, rather than WMS generated maps (images).

The OpenGIS® Geography Markup Language (GML) Encoding Specification provides an Extensible Markup Language (XML) grammar written in XML Schema for the modeling, exchange, and transport, and storage of geographic information. Key concepts are drawn from OpenGIS® Abstract Specifications and the ISO 19100 series of geographic information standards developed by ISO Technical Committee 211.

Web feature services (WFS) – The OpenGIS® Web Feature Service (WFS) Implementation Specification allows a client to retrieve and update geospatial data encoded in Geography Markup Language (GML) from multiple Web Feature Services.

Other Standards Bodies Other standards bodies provide specifications for Web processing. The World Wide Web Consortium (W3C) develops interoperable technologies (specifications, guidelines, software, and tools) to develop the Web to its full potential. It developed Extensible Markup Language (XML) that plays an important role in data exchange on the Web and serves as the basis for the OpenGIS® Geography Markup Language (GML) Encoding Specification. The Organization for the Advancement of Structured Information Standards (OASIS) produces worldwide standards for security, Web services, XML conformance, and other areas that enable electronic business (e-business).

Functional Relationships Among Standards Bodies There are functional relationships among ISO, INCITS L1, OGC, and other standards bodies. The ISO 19100 series of geographic information standards provides components (spatial and topological representation of geographic features, spatial referencing by coordinates and geographic identifiers, feature cataloguing methodology, and metadata) that can be used to develop standards for data themes. INCITS L1 develops standards for data themes based on ISO 19100 series of geographic information standards. OGC develops interoperability specifications that enable Web processing of data based on the ISO 19100 series of geographic information standards; in fact, several standards in the ISO 19100 series of geographic information standards serve as OpenGIS® Abstract Specifications. Finally, other standards bodies such as W3C and OASIS provide the specifications to enable Web processing.

Clearinghouse/Geodata.gov

Sharing of geographic data, maps, and online services is enabled through an online portal, geodata.gov, where users can search metadata held within the NSDI Clearinghouse Network.

The geodata.gov portal provides "one-stop" access to registered geographic information and related online access services within the United States. It serves as a public gateway for access to geospatial information and data. Geographic data, imagery, applications, documents, web sites and other resources have been catalogued for discovery in this portal. The metadata records were submitted to the portal by government agencies, individuals, and companies or harvested from geospatial clearinghouses. Registered map services using OpenGIS® Specifications allow casual users to build online maps using data from many sources. Registered data access and download services also exist for use by those interested in downloading and analyzing the data using GIS or viewer software. Also, organizations can publish data and search for partners for data collection and acquisition.

The NSDI Clearinghouse Network is a community of distributed data providers that publish collections of metadata that describe their map and data resources within their areas of responsibility. Each metadata collection, known as a Clearinghouse Node, is hosted by an organization to publicize the availability of data within the NSDI. The metadata in these nodes are searched by the geodata.gov portal to provide quick assessment of the extent and properties of available geographic resources.

A "digital geospatial data set" is the primary item being described by metadata in the Clearinghouse. The definition of a data set can be adjusted to meet a given agency's requirements, but it generally corresponds to the smallest identifiable data product (e. g. file) for which metadata are customarily collected. This may equate to a specific satellite image or vector data set that is managed by a data producer or distributor. Collections of data sets (e. g. flight lines, satellite "paths", map series, or data series) may also have generalized metadata that could be inherited by individual data sets.

Key Applications

Implementation of NSDI concepts supports sharing of geospatial data among Federal, State, Tribal, and local government agencies, academia, the non-profit sector, the commercial sector, and many other communities.

Future Directions

Geospatial Line of Business

Geospatial Line of Business (Geospatial LoB) is a Presidential Initiative established to identify opportunities for optimizing and consolidating Federal geospatial-related investments to reduce the cost of government and improve services to citizens. The goals of Geospatial LoB are:

- Collaboration for geospatial-related activities and investments across all levels of government and different segments of the geospatial community;
- Optimized and standardized common geospatial functions, services, and processes; and
- Cost efficient acquisition, processing, and access to geospatial data and information.

Desired outcomes of the Geospatial LoB include:

- Clarified performance responsibilities and accountability
- A more collaborative and performance-oriented culture
- Multi-mission delivery capabilities
- More effective investments through increased sharing and reuse
- Better service to agencies and citizens through increased functionality and more coordinated access to geospatial information
- Improved data, services, and tools

Development of North American Profile of ISO 19115, Geographic Information – Metadata

ISO Technical Committee 211 developed ISO 19115, Geographic information – Metadata, which ISO approved and published as an International Standard (IS) in 2005. Each nation can craft their own profile of ISO 19115, with the requirement that it include 13 core elements in ISO 19115. OMB Circular A-119 directs Federal agencies to use voluntary consensus standards, such as those developed by ISO. INCITS L1 is leading the development of a U.S. Profile of ISO 19115. After ANSI has approved the U.S. Profile of ISO 19115, the FGDC will adopt the profile to supersede the Content Standard for Digital Geospatial Metadata (Version 2.0), FGDC-STD-001-1998.

Through an MOU between the Canadian General Standards Board and INCITS, the U.S. and Canada will have common profiles of ISO 19115. There are plans to involve Mexico in the development of a common North American Profile, and there is potential for expansion and coordination with Latin America for a Metadata Profile for the Americas.

Approval of Framework Data Standard as American National Standard

The Framework Data Standard establishes common requirements for data exchange for the seven framework data themes. The Framework Data Standard specifies a minimal level of data content that data producers, consumers, and vendors are expected to use for the interchange of framework data, including through Web services. The draft framework data standards were initially developed through the Geospatial One-Stop e-Government initiative; however, the FGDC assumed leadership for continued standards activities.

INCITS has officially registered this project as Project 1574-D, Information Technology – Geographic Information Framework Data Content Standard. Approval of the Framework Data Standard as an American National Standard is projected in 2007.

Cross References

► Cadastre
► Computing Fitness of Use of Geospatial Datasets
► Data Infrastructure, Spatial
► Geographic Coverage Standards and Services
► Metadata and Interoperability, Geospatial
► OGC's Open Standards for Geospatial Interoperability
► Web Services, Geospatial

Recommended Reading

1. Federal Geographic Data Committee, www.fgdc.gov
   The FGDC website provides extensive resources about the NSDI concepts discussed in this entry
2. Geospatial One Stop (GOS), www.geodata.gov
   The Geospatial One Stop web portal enables users to search a catalog of geospatial information containing thousands of metadata records, view metadata, download data and services, and interact with Web services
3. GSDI Association, www.gsdi.org
   The GSDI Association is an inclusive organization of organizations, agencies, firms, and individuals from around the world. The purpose of the organization is to promote international cooperation and collaboration in support of local, national and international spatial data infrastructure developments that will allow nations to better address social, economic, and environmental issues of pressing importance
4. Spatial Data Infrastructure Cookbook, http://www.gsdi.org/gsdicookbookindex.asp
   This SDI Implementation Guide or Cookbook, through the support of the Global Spatial Data Infrastructure community, provides geographic information providers and users with the necessary background information to evaluate and implement existing components of SDI. It also facilitates participation within a growing (digital) geographic information community known as the Global Spatial Data Infrastructure (GSDI)

Navigation Aid

► Wayfinding, Landmarks

Nearest Neighbor

► Cloaking Algorithms for Location Privacy
► Nearest Neighbors Problem

Nearest Neighbor Algorithm

► Skyline Queries

Nearest Neighbor Queries in Network Databases

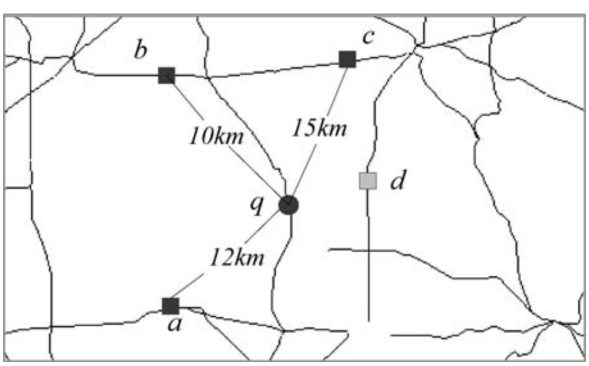
DIMITRIS PAPADIAS[1], MAN LUNG YIU[2], NIKOS MAMOULIS[3], YUFEI TAO[4]
[1] Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China
[2] Department of Computer Science, Aalborg University, Aalborg, Denmark
[3] Department of Computer Science, University of Hong Kong, Hong Kong, China
[4] Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong, China

Synonyms

Spatial network databases; Location based services; Euclidean Restriction

Definition

Spatial network databases (SNDB) are becoming increasingly important since, in practice, objects can usually move only on a pre-defined set of trajectories as specified by the underlying network (road, railway, river etc.). In this case, the important measure is the network distance, i. e., the length of the shortest trajectory connecting two objects, rather than their Euclidean distance. Every spatial query type (e. g., *nearest neighbors*, *range search*, *spatial joins*, etc.) has a counterpart in SNDB. Consider, for instance, the road network of Fig. 1, where the rectangles correspond to hotels. If a user at location q poses the query "find the nearest hotel", the result is b (the numbers in the figure correspond to network distance). Note that the Euclidean nearest neighbor is d, which is actually the farthest hotel in the network.



Nearest Neighbor Queries in Network Databases, Figure 1 Road network query example

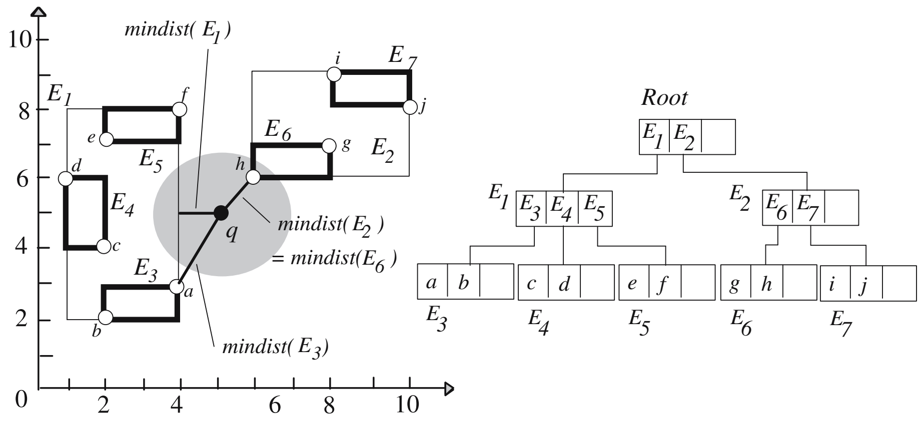
A crucial pre-requisite for solving these queries in SNDB is a realistic architecture, which captures *spatial entities* (e. g., hotels) and the underlying network, preserving both Cartesian co-ordinates and connectivity. In addition, this architecture must take into account real-life constraints, such as unidirectional roads, "off-network" (but still reachable) entities, etc. Furthermore, although the network is almost static, the entities may change with relatively high frequencies (e. g., a new/existing hotel opens/closes). It is also possible that entire entity sets are added as more information or services become available (e. g., a new restaurant dataset is incorporated in the system).

Historical Background

R-trees [1] are the most popular indexes for Euclidean query processing due to their simplicity and efficiency. The R-tree can be viewed as a multi-dimensional extension of the B-tree. Figure 2 shows an exemplary R-tree for a set of points $\{a,b,\ldots,j\}$ assuming a capacity of three entries per node. Points that are close in space (e. g., a,b) are clustered in the same leaf node (E_3) represented as a minimum bounding rectangle (MBR). Nodes are then recursively grouped together following the same principle until the top level, which consists of a single root.

A *nearest neighbor* (NN) *query* retrieves the ($k \geq 1$) data point(s) closest to a query point q. The R-tree NN algorithm of [4] keeps a *heap* with the entries of the nodes visited so far. Initially, the heap contains the entries of the root sorted according to their minimum distance (*mindist*) from q. The entry with the minimum *mindist* in the heap (E_1 in Fig. 2) is expanded, i. e., it is removed from the heap and its children (E_3, E_4, E_5) are added together with their *mindist*. The next entry visited is E_2 (its *mindist* is currently the minimum in the heap), followed by E_6, where the actual result (h) is found and the algorithm terminates, because the *mindist* of all entries in the heap is greater than the distance of h. The algorithm can be easily extended for the retrieval of k nearest neighbors (kNN). Furthermore, it is optimal (it visits only the nodes necessary for obtaining the nearest neighbors) and *incremental*, i. e., it reports neighbors in ascending order of their distance to the query point, and can be applied when the number k of nearest neighbors to be retrieved is not known in advance.

R-trees are not sufficient for indexing SNDB, which are better captured by disk-based graph structures. A graph is usually represented either as a 2D matrix (where each entry corresponds to an edge between a pair of nodes), or an adjacency list. Adjacency lists are preferable for applications, such as road networks, where the graphs are sparse. The main issue for adapting this representation to sec-



Nearest Neighbor Queries in Network Databases, Figure 2
An R-tree example and nearest neighbor query

ondary memory is how to cluster lists of adjacent nodes in the same disk page, in order to take advantage of the access locality and minimize the I/O. The *connectivity-clustered access method* (CCAM) [12] generates a single-dimensional ordering of the nodes (using Z-ordering) and stores the lists of neighbor nodes together.

A common problem in spatial graphs refers to computation of the *shortest path* connecting two points. The most popular shortest path algorithm, proposed by Dijkstra [3], starts from the source and expands the route towards the destination, using a priority queue to store visited nodes (sorted according to their distance from the source node). Several variants of this algorithm differ on how they manage the priority queue. Materialization techniques accelerate shortest path processing (at the expense of space requirements) by using pre-computed results stored in materialized views [7].

CCAM and similar architectures only preserve connectivity (but not location) information. Thus, they are applicable only to shortest paths and other graph traversal algorithms, but not to conventional spatial query processing. Next we present a architecture that can be used for several spatial queries, and describe algorithms for processing NN search.

Scientific Fundamentals

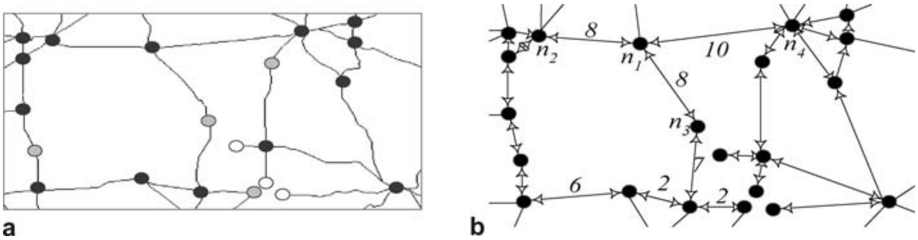
We assume a digitization process that generates a *modeling graph* from an input spatial network. Considering the road network in the introduction, the graph nodes generated by this process are: (i) the network junctions (e. g., the black points in Fig. 3a), (ii) the starting/ending point of a road segment (white), and (iii) depending on the application, additional points (gray) such as the ones where the curvature or speed limit changes. The graph edges preserve the connectivity in the original network. Figure 3b shows the (modeling) graph for the network of Fig. 3a; nodes at the boundary of the data space and the network distance of most edges are omitted for clarity.

In the sequel we use the term edge/segment to denote a direct link in the graph/network. Each edge connecting nodes n_i, n_j stores the *network distance* $d_N(n_i,n_j)$. For nodes that are not directly connected, $d_N(n_i,n_j)$ equals the length of the shortest path from n_i to n_j. If unidirectional traffic is allowed (e. g., one-way road segments), $d_N(n_i,n_j)$ is asymmetric (i. e., it is possible $d_N(n_i,n_j) \neq d_N(n_j,n_i)$). Furthermore, $d_E(n_i,n_j) \leq d_N(n_i,n_j)$, i. e., the corresponding Euclidean distance $d_E(n_i,n_j)$ *lower bounds* $d_N(n_i,n_j)$ (equality holds only if n_i, n_j are connected by a straight segment). We refer to this fact as the *Euclidean lower-bound property*.

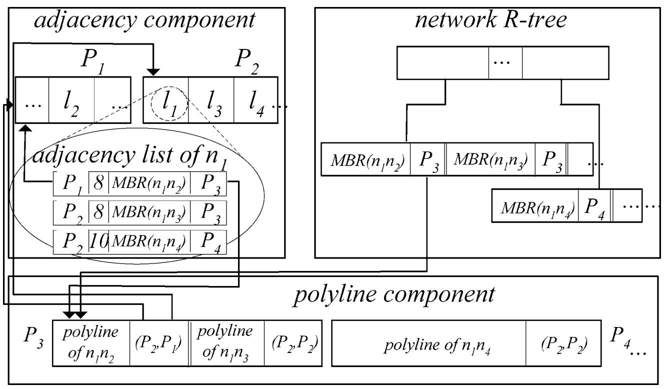
Constraints, such as special traffic controls, can be modeled by including extra nodes to the graph. In order to simplify the presentation, we describe the architecture for the basic functionality, where nodes have identical types and edges only store network distance. We separate the spatial entities (e. g., hotels) from the underlying network, by indexing each entity dataset using an R-tree. This division has many advantages: (i) all conventional (Euclidean) queries, which do not require the network, can be efficiently processed by the R-trees, (ii) queries combining network and Euclidean aspects are supported, (iii) dynamic updates in each dataset are handled independently, (iv) new/existing datasets can be added to/removed from the system easily, and (v) specific optimizations can be applied to each individual (network or entity) dataset.

The network storage scheme consists of three components. The *adjacency component* captures the network connectivity. The adjacency lists of the nodes close in space, according to their Hilbert values, are placed in the same disk page. In Fig. 4 (based on Fig. 3b), the list l_1 of n_1 consists of 3 entries, one for each of its connected nodes (n_2, n_3, n_4) (ignoring nodes outside the boundary). The first entry (for edge n_1n_2) has the form ⟨NBptr(n_2), 8, MBR(n_1n_2),

N



Nearest Neighbor Queries in Network Databases, Figure 3 Graph modeling of the road network **a** a road network, **b** the modeling graph



Nearest Neighbor Queries in Network Databases, Figure 4 Example of the proposed architecture

PLptr(n_1n_2)⟩, where NBptr(n_2) points to the disk page (i. e., P_1) containing the adjacency list l_2 of n_2. NBptr(n_2) enables fast access to the neighboring node n_2, without any additional look-up. The next field (8) is the network distance of edge n_1n_2. MBR(n_1n_2) records the minimum bounding rectangle of the actual poly-line n_1n_2 in the original network, which is stored in the disk page ($=P_3$) specified by PLptr(n_1n_2). The other adjacency entries (for n_3, n_4) have the same format. The *poly-line component*, stores the detailed poly-line representation of each segment in the network. A poly-line entry n_in_j also includes a pair of pointers to the disk pages containing the adjacency lists of its endpoints n_i, n_j. The last component is a *network R-tree* that indexes the MBRs of the poly-lines and supports queries exploring the spatial properties of the network. Each leaf entry contains a pointer to the disk page storing the corresponding poly-line.

The architecture supports the following primitive operations for SNDB:

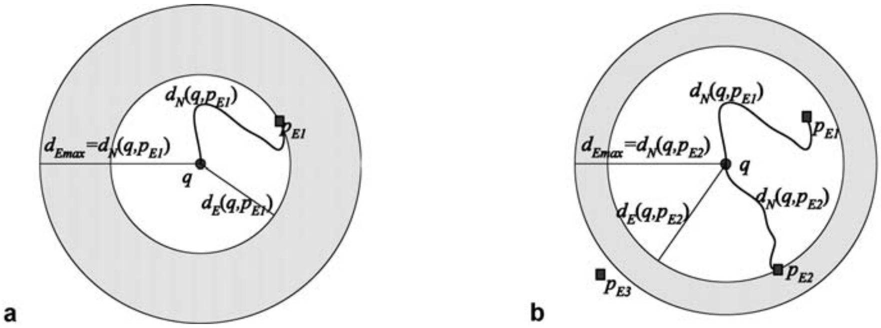
(i) *check_entity*(*seg*, *p*) is a Boolean function that returns true if point (entity) *p* lies on the network segment *seg* (we say that *seg covers p*). In accordance with the conventional spatial databases methodology, the MBR of *seg* is used for filtering and its poly-line representation for refinement.

(ii) *find_segment*(*p*): outputs the segment that covers point *p* by performing a point location query on the network R-tree. If multiple segments cover *p*, the first one found is returned. This function is applied whenever a query is issued, to locate the segment on which the query point lies.

(iii) *find_entities*(*seg*): returns entities covered by segment *seg*. Specifically, it first finds all the candidate entities that lie in the MBR of *seg*, and then eliminates the false hits using the poly-line of *seg*.

(iv) *compute_ND*(p_1,p_2): returns the network distance $d_N(p_1,p_2)$ of two arbitrary points p_1, p_2 in the network, by applying a (secondary-memory) algorithm to compute the shortest path from p_1 to p_2. Network distance computations can be facilitated by materialization of pre-computed results (e. g., [7]).

Using this architecture, we present two frameworks, *Euclidean restriction* and *network expansion*, for processing nearest neighbors queries. Assuming that only one NN is required, the Incremental Euclidean Restriction (IER) algorithm first retrieves the Euclidean nearest neighbor p_{E1} of q, using an incremental kNN algorithm (e. g., [4]) on the entity R-tree. Then, the network distance $d_N(q,p_{E1})$ of p_{E1} is computed (by *compute_ND*(q,p_{E1})). Due to the Euclidean lower-bound property, objects closer (to q) than p_{E1} in the network, should be within Euclidean distance $d_{Emax} = d_N(q,p_{E1})$ from q, i. e., they should lie in the shaded area of Fig. 5a. In Fig. 5b, the second Euclidean NN p_{E2} is then retrieved (within the d_{Emax} range). Since $d_N(q,p_{E2}) < d_N(q,p_{E1})$, p_{E2} becomes the current NN and d_{Emax} is updated to $d_N(q,p_{E2})$, after which the search region (for potential results) becomes smaller (the shaded area in Fig. 5b).



Nearest Neighbor Queries in Network Databases, Figure 5 Finding the NN p_{E2} **a** 1st Euclidean NN, **b** 2nd Euclidean NN

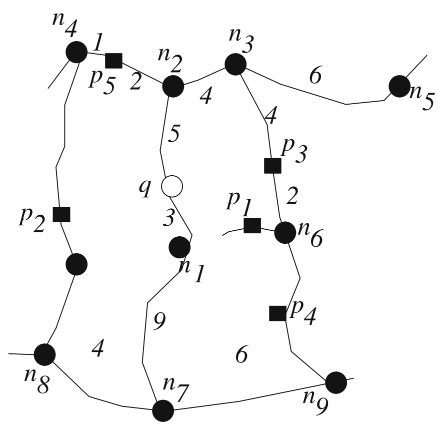
Since the next Euclidean NN p_{E3} falls out of the search region, the algorithm terminates with p_{E2} as the final result.

The extension to k nearest neighbors is straightforward. The k Euclidean NNs are first obtained using the entity R-tree, sorted in ascending order of their network distance to q, and d_{Emax} is set to the distance of the kth point. Similar to the single NN case, the subsequent Euclidean neighbors are retrieved incrementally, while maintaining the k (network) NNs and d_{Emax} (except that d_{Emax} equals the network distance of the kth neighbor), until the next Euclidean NN has larger Euclidean distance than d_{Emax}.

IER performs well if the ranking of the data points by their Euclidean distance is similar to that with respect to the network distance. Otherwise, a large number of Euclidean NNs may be inspected before the network NN is found. Figure 6 shows an example where the black points represent the nodes in the modeling graph and rectangles denote entities. The nearest entity to the query q (white point) is p_5. The subscripts of the entities (p_1,p_2,... ,p_5) are in ascending order of their Euclidean distance to q. Since p_5 has the largest Euclidean distance, it will be examined after all other entities, i. e., p_1 to p_4 correspond to *false hits*, for which the network distance computations are redundant.

To remedy this problem, the Incremental Network Expansion (INE) algorithm performs network expansion (starting from q), and examines entities in the order they are encountered. Specifically, INE first locates the segment n_1n_2 that covers q, and retrieves all entities on n_1n_2 (using the primitive operation *find_entities*). Since no point is covered by n_1n_2, the node (n_1) closest to the query is expanded (while, the second endpoint n_2 of n_1n_2 is placed in a queue Q). No data point is found in n_1n_7 and n_7 is inserted to $Q = \langle (n_2,5), (n_7,12) \rangle$. The expansion of n_2 reaches n_4 and n_3, after which $Q = \langle (n_4,8), (n_3,9), (n_7,12) \rangle$ and point p_5 is discovered on n_2n_4 (while no point is found on n_2n_3). The distance $d_N(q,p_5) = 7$ provides a bound d_{Nmax} to restrict the search space. The algorithm terminates now since the next entry n_4 in Q has larger distance (i. e., 8) than d_{Nmax}.

As shown in [11] both *Euclidean restriction* and *network expansion* can be applied to a variety of spatial queries, including range search, spatial joins and closest-pairs. *Euclidean restriction* provides an intuitive way to deal with spatial constraints. If for instance, we want to "find the two nearest hotels to the south", we only need to retrieve the Euclidean neighbors in the area of interest. On the other hand, although *network expansion* is still applicable, it has limited pruning power on queries with selective spatial conditions. Considering again the example query, the network should be also expanded to the north of the query point, because subsequent nodes may lead to a nearest neighbor to the south. Furthermore, the *Euclidean restriction* framework assumes the lower bounding property, which may not always hold in practice. If, for instance, the edge cost is defined as the expected travel time, the Euclidean distance cannot confine the search space (unless we make additional assumptions, such as maximum speed). On the contrary, *network expansion* permits a wide variety of costs associated with the edges. It assumes, however, that the cost increases mono-



Nearest Neighbor Queries in Network Databases, Figure 6 Finding the NN p_5

tonically with the path (i. e., a path cannot be cheaper than one of its sub-sets), because, otherwise there is no bound in the expansion process. Dijkstra's algorithm is also based on the same assumption, which is realistic for all SNDB applications.

Key Applications

Geographic Information Systems

Spatial (e. g., road, river) networks are common in Geographic Information Systems (GIS). Efficient query mechanisms are important for dealing with the large and ever increasing amount of spatial data in several GIS applications.

Navigation Systems

Navigation systems are already incorporated in vehicles and mobile devices (e. g., PDAs). Flexible SNDB architectures and advanced algorithms will provide the means for enhanced location-based services (in addition to the common shortest path services currently offered by most systems).

Decision Support

Nearest neighbor and related queries are sometimes important components for decision support systems. For example, assume a supermarket chain that wants to build a number of warehouses close to existing sale facilities. This task is related to NN search and its solution could invoke SNDB architectures and algorithms.

Future Directions

Several specialized techniques can be used to speed up evaluation of NN queries on road networks. [8] uses the concept of network Voronoi cells and materialization. [6] presents the *island* approach for balancing query and materialization costs. In another direction, [5] proposes tree-based structures for capturing shortest paths in a road network and develops techniques for fast processing of NN queries on those structures. [13] addresses continuous nearest neighbor (CNN) queries that retrieve all nearest neighbors along a given route. [9] extends the Voronoi-based solution of [8] for computing CNN queries, while [2] combines network expansion and precomputed NN lists of network nodes for deriving query results. Other directions for future work include maintenance of long running NN queries on moving objects [10] and alternative forms of NN search (such as aggregate NN [14] and reverse NN [15]).

Cross References

► Nearest Neighbor Query
► R*-tree
► Trip Planning Queries in Road Network Databases

Recommended Reading

1. Beckmann, N., Kriegel, H.P., Schneider, R., Seeger, B.: The R*-tree: An Efficient and Robust Access Method for Points and Rectangles. SIGMOD (1990)
2. Cho, H.J., Chung, C.W.: An Efficient and Scalable Approach to CNN Queries in a Road Network. VLDB (2005)
3. Dijkstra, E.W.: A Note on Two Problems in Connection with Graphs. Numeriche Mathematik, vol. 1, 269–271 (1959)
4. Hjaltason, G., Samet, H.: Distance Browsing in Spatial Databases. TODS **24**(2), 265–318 (1999)
5. Hu, H., Lee, D.L., Xu, J. Fast Nearest Neighbor Search on Road Networks. EDBT (2006)
6. Huang, X., Jensen, C. S., Saltenis, S.: The Islands Approach to Nearest Neighbor Querying in Spatial Networks. SSTD (2005)
7. Jing, N., Huang, Y.W., Rundensteiner, E.A.: Hierarchical Encoded Path Views for Path Query Processing: an Optimal Model and its Performance Evaluation. TKDE **10**(3), 409–432 (1998)
8. Kolahdouzan, M., Shahabi, C.: Voronoi-Based K Nearest Neighbor Search for Spatial Network Databases. VLDB (2004)
9. Kolahdouzan, M., Shahabi, C.: Continuous K-Nearest Neighbor Queries in Spatial Network Databases. STDBM (2004)
10. Mouratidis, K., Yiu, M.L., Papadias, D., Mamoulis, N.: Continuous Nearest Neighbor Monitoring in Road Networks. VLDB (2006)
11. Papadias, D., Zhang, J., Mamoulis, N., Tao, Y.: Query Processing in Spatial Network Databases. VLDB (2003)
12. Shekhar, S., Liu, D.: CCAM: A Connectivity-Clustered Access Method for Networks and Network Computations. TKDE **19**(1), 102–119 (1997)
13. Shekhar, S., Yoo, J.S.: Processing In-Route Nearest Neighbor Queries: A Comparison of Alternative Approaches. ACM GIS (2003)
14. Yiu, M.L., Mamoulis, N, Papadias, D.: Aggregate Nearest Neighbor Queries in Road Networks. TKDE **17**(6), 820–833 (2005)
15. Yiu, M.L., Papadias, D., Mamoulis, N., Tao, Y.: Reverse Nearest Neighbors in Large Graphs. TKDE **18**(4), 540–553 (2006)

Nearest Neighbor Query

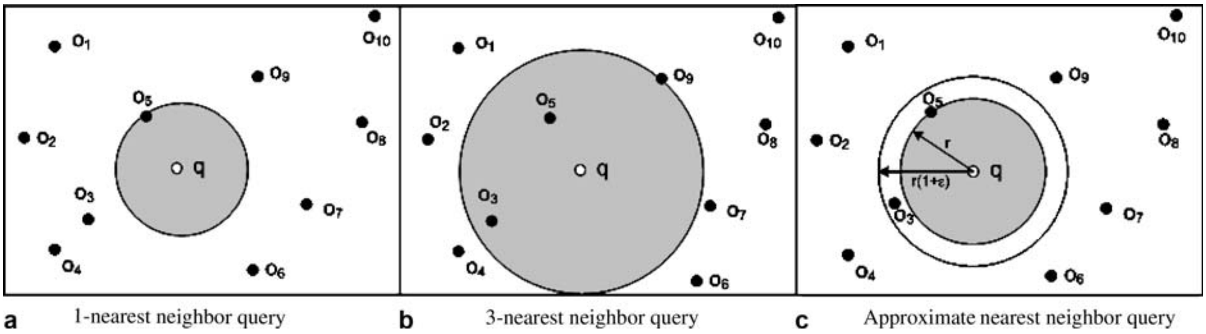
FENG CHEN, CHANG-TIEN LU
Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA

Synonyms

K-nearest neighbor query; Point nearest-neighbor query; Closest point query

Definition

Given a set P of n points in a d-dimensional space $\Re^d$ and a query point $q \in \Re^d$, a nearest neighbor (*NN*) query is



Nearest Neighbor Query, Figure 1 Three examples of nearest neighbor query

to find the set of nearest points to the query point q. Formally, $NN(q) = \{p|p \in P, \forall o \in P, (o \neq p), |qp| < |qo|\}$. In most cases, $\Re^d$ refers to a d-dimensional Euclidean space. Figure 1a shows an example of *NN* query, where o_5 is the closest point to the query point q.

There are two major variants of *NN* queries. One is k-nearest neighbor (k-*NN*) query, which retrieves the k nearest points to the query point q. Another is approximate nearest neighbor query [1], which takes an error parameter ε and finds a point within P whose distance to the query point q is at most $(1+\varepsilon)$ times the distance to the true nearest neighbor. For example, in Fig. 1b, the three nearest neighbor points of the query point q are o_5, o_3, and o_9, sorted by the descending value of the distances to q. As illustrated in Fig. 1c, the approximate nearest neighbor of the query point q is point o_3, whose distance to q is smaller than $(1+\varepsilon) \cdot r$, where r refers to the distance from the true nearest neighbor o_5 to q.

In addition to the exact and approximate *kNN* queries, many other *NN* variants have been proposed in recent years, such as range nearest neighbors and nearest neighbors for extended objects. A Range Nearest Neighbor (*RNN*) query retrieves the nearest neighbor for every point in a given range, which is usually assumed as a (hyper) rectangle [2]. A nearest neighbor query for extended objects assumes each object as a polygon and finds the nearest neighbor object to a query object [3]. The distance between two extended objects can be defined as the minimal distance between these two objects.

Historical Background

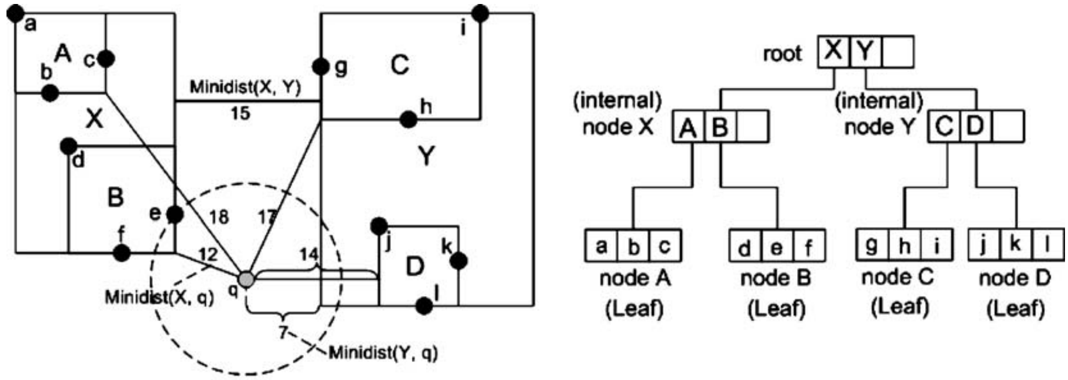
Nearest neighbor search is one of the oldest problems in computer science. This problem was first introduced by E. Fix and J. Hodges [4,5]. The *NN* algorithm was one of the first algorithms used to find a solution to the traveling salesman problem (*TSP*) [6]. In 1977, Rosenkrantz et al. showed that the *NN* algorithm can be used as an efficient approach to approximate the travel salesman problem [7]. Later, several algorithms and theoretical performance bounds were devised for exact and approximate *NN* processing [8,9].

NN query first appeared in Geographic Information System (*GIS*) around 1990s and has now become a frequently encountered query type. Several efficient spatial access methods were discussed in [10], but still very few were applied to the *NN* query. In [11], several heuristics were first proposed to process the *NN* query based on quadtrees, and were later refined in [8]. The first *NN* algorithm for *R*-trees, called depth-first search (*DFS*), was proposed in 1995 by Roussopoulos et al. [3]. The *DFS* algorithm is suboptimal because it accesses more nodes than necessary. An I/O optimal algorithm called best-first search (*BFS*) was later introduced in 1999 by Hjaltason and Samet [12]. Although these algorithms were designed in the context of *R*-tree, they are applicable to other spatial structures. Up to now, *DFS* and *BFS* are the two most popular branch-and-bound paradigms for the *NN* query processing.

N

Scientific Fundamentals

The performance of *NN* search can be measured by the space and time complexities of the search algorithms. In recent years, a number of solutions have been proposed for the *NN* search problem. Based on the different index structures, these algorithms can be classified into three basic categories, including linear search, space partitioning, and locality sensitive hashing (*LSH*). Linear search is to compute the distance between the query point and each point in the data set, and then return the point with the minimum distance. This solution is straightforward to implement but tends to be intractable when either the number of dimension or the size of the data set increases. Its time complexity is $O(N \cdot d)$, where N and d refer to the cardinality of the data set and the number of dimensions, respectively. Because it does not require any index structure, the space



a The object and node placement **b** The corresponding R-tree

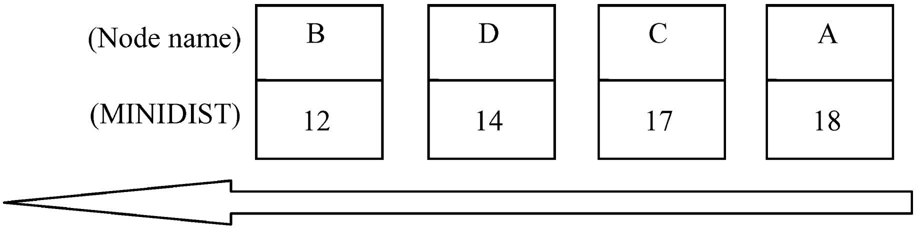
Nearest Neighbor Query, Figure 2 An example of a NN query on an R-tree index

complexity is the storage of the data set. Locality sensitive hashing (and its variants) is an effective solution for approximate the *NN* search problem in high dimensions based on hashing index structures. The basic idea is to hash the points based on several hash functions so that, for each function, it is ensured that points closer to each other have a higher probability to be stored in a same bucket. As a result, given a query point, it is only necessary to search the points stored in the bucket containing the query point. *LSH* was originally proposed for the case when the points live in binary Hamming space [13]. Later the method was extended to handle the case when the points are in Euclidean space, followed by several improvements in performance [14,15]. The latest version of *LSH* solutions [15] achieves query time of $O(d\,n^{1/c^2+o(1)})$ and space of $O(d\,n + n^{1+1/c^2+o(1)})$, where n denotes the cardinality of the data set, d refers to the number of dimensions, and c is an approximation factor.

Space partitioning method first indexes the set of points by a spatial access method (e. g., *kd*-tree [16], *R*-tree [17]) and then processes an *NN* query based on two interleaving phases: the first phase, called filtering, is to use branch and bound technique to efficiently prune distant index nodes, and the second phase, called refinement, is the in-memory computation of the nearest neighbors. Depending on the storage of the index structure, the algorithms can be categorized into two sub-categories, namely memory-based and secondary-storage-based algorithms. *Quad*-trees [18] and *kd*-trees are two of the most popular index structures for memory-based algorithms. As for the second category, because *R*-tree (and its variants) is a natural generalization of *B*-tree and has a balanced structure that clusters objects by their spatial proximity, it is the most popular spatial index structure. The time complexity contains two parts: *I/O* cost and *CPU* cost. Because *I/O* cost is normally much higher than *CPU* cost, the existing algorithms mostly focus on the first phase to minimize *I/O* operations. The two popular *NN* algorithms, *DFS* and *BFS*, are described as follows to show how to process an *NN* query.

Figure 2 shows an *R*-tree with a node capacity of three for a point set $P = \{a, b, c, d, e, f, g, h, i, j, k, l\}$. Figure 2a shows the placements, containments, and overlapping relationships of the objects and index nodes, and Fig. 2b shows the structure of the corresponding *R*-tree, including the root, intermediate nodes X and Y, and leaf nodes A, B, C, and D. Every internal node in the *R*-tree stores a minimum bounding rectangle (*MBR*) that spatially contains the rectangles in its child nodes. The points that are close to each other are clustered in one leaf node and recursively the nodes that are close to each other are grouped into upper level nodes till the root node is finally generated. The minimal distance between the query point q and an internal node N, named $MINIDIST(q, N)$, is defined as the minimal distance between q and the node N's *MBR*. The minimal distance between two internal nodes N_1 and N_2, named $MINIDIST(N_1, N_2)$, is defined as the minimal distance between the *MBRs* of the nodes N_1 and N_2. Figure 2a shows the minimal distance between query point q and internal node X, and that between internal node X and internal node Y.

The first *NN* algorithm for *R*-trees [3], named depth-first search (*DFS*), recursively visits index nodes to search for nearest neighbor candidates. More specifically, index nodes are accessed based on the depth-first search manner. When an index node is accessed, its child nodes are accessed with a certain order that is determined by sorting the child nodes based on their *MINIDIST* distances to the query point. When a leaf node is accessed, the points inside the node are retrieved and the *NN* candidates are updated. For example, in Fig. 2, *DFS* first accesses the



Nearest Neighbor Query, Figure 3 An example of priority queue

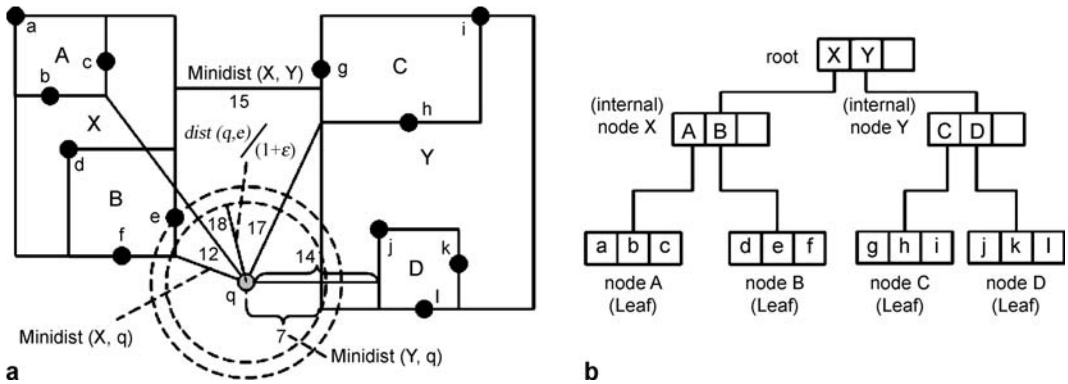
root, followed by nodes *Y* and *D*, since *MINIDIST*(*Y*, *q*) < *MINIDIST*(*X*, *q*) and *MINIDIST*(*D*, *q*) < *MINIDIST*(*C*, *q*). Because node *D* is a leaf node, the points (*j*, *k*, *l*) inside node *D* are retrieved and the candidate point is updated as point *j*, since *j* has the smallest distance to *q* than those of other points (*l*, *k*) inside *D*. During backtracking to the upper level (node *Y*), the node whose *MINIDIST* distance to the query point is larger than the best distance (*best_dist*), the distance from the candidate point to the query point, will be pruned. In the example, because *MINIDIST*(*C*, *q*) is larger than the best distance, *MINIDIST*(*j*, *q*), node *C* is pruned without visit. Later, after backtracking to the upper level (root), node *X* is accessed, followed by node *B*. Because node *B* is a leaf node, its points (*d*, *e*, *f*) are retrieved and, because *MINIDIST*(*e*, *q*) is smaller than *MINIDIST*(*j*, *q*), the candidate point is updated as point *e*. During backtracking to the upper level (node *X*), node *A* is pruned without visit, since *MINIDIST*(*A*, *q*) is larger than *best_dist* $|eq|$. Finally, the candidate point *e* is returned as the actual *NN* point.

The preceding *DFS* algorithm can be easily extended to the retrieval of $k > 1$ nearest neighbors: the candidate set will preserve and maintain *k* candidate points in an ordered list of *k* pairs $\langle p, |pq|\rangle$, which is descendingly sorted based on $|pq|$, and *best_dist* is the distance between the *k*-th nearest neighbor and the query point. Whenever a new candidate point is found, the element of the candidate list will be replaced with the new candidate point, and the candidate list will be re-ordered. An example of *2NN* query is described as follows: first, *DFS* accesses the root followed by node *Y* and then node *D*. Because node *D* is a leaf node, the points inside the node are retrieved, then the candidate set is updated as $\{\langle j, |qj|\rangle, \langle l, |ql|\rangle\}$, and the *best_dist* is set as $|ql|$, which is the distance from the second candidate point *l* to the query point *q*. During backtracking to the upper level (node *Y*), because *MINIDIST*(*C*, *q*) is larger than *best_dist*, node *C* is pruned without visit. Later, after backtracking to the upper level (root), node *X* is accessed, followed by node *B*. Because the node *B* is a leaf node, its points are retrieved and ordered based on their *MINIDIST* distances. These points are processed according to the sorted sequence. More specifically, the point *e* is first processed. Because $|eq|$ is smaller than *best_dist*, it is regarded as the new candidate point. The candidate set and *best_dist* are updated as $\{\langle e, |qe|\rangle, \langle j, |qj|\rangle\}$ and $|qj|$, respectively. Then point *f* is processed. Because $|qf|$ is larger than *best_dist*, it is pruned. The next processed point is *d*. Because $|dq|$ is larger than *best_dist*, it is also pruned. After the node *B* is processed, it is backtracked to the upper level (node *X*), and node *A* is pruned without visit because *MINIDIST*(*A*, *q*) is larger than *best_dist*. Finally, the *2NN* points (*e*, *j*) of the query point *q* are returned.

The *DFS* algorithm is suboptimal because it accesses some unnecessary nodes. As proven by Berchtold et al. [19], an optimal algorithm should visit only nodes which intersect with the *NN*-sphere (also called search region, or vicinity circle). The *NN*-sphere of a query point *q* is defined as the sphere with center *q* and radius *r*, the distance of the nearest neighbor and the query point. For example, in Fig. 2, the dotted circle region refers to the *NN*-sphere. Because nodes *Y*, *X*, and *B* intersect with the *NN*-sphere, an optimal algorithm should visit only these three nodes, whereas *DFS* algorithm accesses an unnecessary node *D*.

The best-first search (*BFS*) algorithm proposed by Hjatason and Samet [12] achieves the optimal I/O performance by using a priority queue to store entries to be explored during the search. The entries, which can be internal nodes, leaf nodes, or points, are sorted based on their *MINIDIST* distances and are processed sequentially. For example, in Fig. 3, $\langle(B, 12), (D, 14), (C, 17), (A, 18)\rangle$ is a priority queue, where the nodes are sorted (from left to right) based their *MINIDIST* distances to the query point *q*. Initially, the priority queue only contains the root node. In each iteration, the head entry of the queue, which has the smallest *MINIDIST* distance, will be popped up. If the entry is an internal node, its child nodes will be pushed into the queue; if the entry is a leaf node, the points inside the node will be pushed into the queue; otherwise, the entry (point) will be returned as the nearest neighbor. For example, in Fig. 2, *BFS* first pushes the root into the priority queue. After the root is popped, *BFS* pushes its two child nodes *X* and *Y* into the queue. Then node *Y* (the head entry in the queue) is popped and its child nodes *C* and *D* are pushed into the queue. The priority queue is updated as $\langle(X, 12), (D, 14), (C, 17)\rangle$. Then node *X* is popped and its child nodes *A* and *B* are pushed into the queue, which is

N



Nearest Neighbor Query, Figure 4 An example of a $(1+\varepsilon)$-Approximate NN query on an R-tree index. **a** The object and node placement. **b** The corresponding R-tree

updated as $\langle(B, 12), (D, 14), (C, 17), (A, 18)\rangle$. Then node B is popped and, because it is a leaf node, the points (e, f, d) inside B are pushed into the queue: $\langle(e, 12.5), (f, 13), (D, 14), (d, 15), (C, 17), (A, 18)\rangle$. After that, the point e is popped and stored as the current nearest neighbor. At this time, *BFS* terminates since the *MINIDIST* distance of the head entry (node D) in the queue is greater than that of the candidate point e, and outputs the point e as the *NN* of the query point q. Hence, unlike *DFS*, node D is not visited.

Similar to *DFS*, *BFS* can be extended to process *kNN* queries ($k > 1$). In addition, it can incrementally output the k-nearest neighbors in a descending order of their distances to the query point. An example of *2NN* query is described as follows: first, *BFS* pushes the root into the priority queue: $\langle(root, 0)\rangle$. After the root is popped, *BFS* pushes two child nodes X and Y into the queue: $\langle(Y, 10), (X, 12)\rangle$. Then node Y is popped and its child nodes C and D are pushed into the queue: $\langle(X, 12), (D, 14), (C, 17)\rangle$. Then node X is popped and its child nodes A and B are pushed into the queue: $\langle(B, 12), (D, 14), (C, 17), (A, 18)\rangle$. Then node B is popped and, because it is a leaf node, the points (e, f, d) inside the node are pushed into the queue: $\langle(e, 12.5), (f, 13), (D, 14), (d, 15), (C, 17), (A, 18)\rangle$. Next the point e is popped and reported as the first nearest neighbor. After that, the point f is popped and reported as the second nearest neighbor. At this time, the algorithm terminates since the 2-nearest neighbors have already been found.

The exact *NN* algorithms as discussed above work well for points in a low-dimensional space. However, their performance deteriorates as the number of dimensions increases because their time complexities are exponentially increased in the number of dimensions. One practical solution for this case is the approximate nearest neighbor search, which can achieve polynomial time complexity under a specified error bound (ε). More specifically, given a set D of data points in a d-dimensional space $\Re^d$, a query point $q \in \Re^d$, and an error bound $\varepsilon > 0$, suppose a point $p\prime \in D$ is the true nearest neighbor of q, then a point $p \in D$ that satisfies the condition of $dist(p, q) < q(1+\varepsilon) \cdot (dist)(p\prime, q)$ is called the $(1+\varepsilon)$-approximate nearest neighbor of q [1].

The algorithm for approximate nearest neighbor search is illustrated as follows using the same example. Suppose the query point is q with an error bound $\varepsilon = 0.2$. A priority queue is used to store entries to be explored during the search, which can be internal nodes or leaf nodes. The entries in the queue are processed iteratively. In each iteration, the head entry of the queue, which has the smallest *MINIDIST* distance, will be popped up. If the entry is an internal node, its child nodes will be pushed into the queue. If the entry is a leaf node, the points inside the node will be visited and the closest point seen so far will be stored as the candidate point. Once a candidate point p is found, the termination distance is updated as $dist(q, p)/(1+\varepsilon)$. That means, as soon as the *MINIDIST* of the current entry exceeds $dist(q, p)/(1+\varepsilon)$, the algorithm will terminate and return the current candidate as the final result. For example, in Fig. 4, it first pushes the root into the priority queue: $\langle(root, 0)\rangle$. After the root is popped, it pushes two child nodes X and Y into the queue: $\langle(Y, 10), (X, 12)\rangle$. Then node Y (the head entry in the queue) is popped and its child nodes C and D are pushed into the queue: $\langle(X, 12), (D, 14), (C, 17)\rangle$. Then node X is popped and its child nodes A and B are pushed into the queue: $\langle(B, 12), (D, 14), (C, 17), (A, 18)\rangle$. Then node B is popped and point e is stored as the candidate point, and the termination distance is updated as the value $dist(q, e)/(1+\varepsilon)$. The priority queue is updated as $\langle(D, 14), (C, 17), (A, 18)\rangle$. Because the current entry (D) has a higher *MINIDIST* to q than the termination distance, the algorithm terminates and returns the current candidate point e as the final result. Note that, in this example, the number of nodes accessed is the same as that of *BFS*. However, usually the number would be lower, because its search region is smaller than that of *BFS*. Similar to *DFS* and *BFS*, the preceding algorithm can be easily generalized to process $(1+\varepsilon)$-approximate k-nearest neighbor queries. Although a number of different algorithms have been proposed for approximate nearest neighbor search, they are similar to the basic algorithm as discussed above except that the index structure used may be different. In addition to the *R*-trees, *kd*-trees and balanced box-decomposition (*BBD*) trees can also be applied to search nearest neighbors. Especially, the *BBD*-based algorithm is the most popular one. This algorithm is possible to preprocess a set of n points in $\Re^d$in $O(dn \log n)$ time and $O(dn)$space, so that given a query point q, and $\varepsilon > 0$, a $(1+\varepsilon)$-approximate *NN* query can be computed in $O\left(c_{d,\varepsilon} \log n\right)$ time, where $c_{d,\varepsilon} < d\lceil 1 + 6d/\varepsilon \rceil^d$is a factor depending on the number of dimensions d and the error parameter ε.

Key Applications

Nearest neighbor search is an important problem in a variety of application domains, including geographic information systems (*GIS*), pattern recognition and classification, multimedia databases, and document retrieval.

Geographic Information System (*GIS*)

Nearest neighbor is one of the most frequently used queries in *GIS* domains. For example, a user may point to a specific location and request the system to find the nearest restaurant. In another situation, a user may point to a specific location, specify a spatial range (e. g., a city district), and request the system to search the nearest hospital that locates inside the specified spatial range. Another more complex query is to find a set of spatial locations that the aggregated travel distance used to visit all of these spatial locations is minimized. For example, a city may have several malls, banks, and parks, and a user may request the system to find a sequence of specific mall, bank, and park, such that it takes the minimum travel time to visit one on each category.

Pattern Recognition and Classification

Pattern recognition and classification is a special field generating strong interest in image analysis and computer vision. Nearest neighbor techniques have been applied in this field to improve the recognition performance. For example, a neural network for pattern classification is typically designed by first constructing the network and then applying a training procedure that compute the boundaries of the decision regions. The nearest neighbor classification has been successfully applied as an alternative technique for pattern classification [20].

Clustering and Outlier Detection

Clustering is to partition a dataset into several groups such that the points within each group are closer to each other than to the points in other groups. Outlier detection is to find a small set of “abnormal” points that are distant from the rest of data. In other words, it is to detect the points that do not belong to each cluster and also cannot be regarded as clusters themselves because of their small size. Nearest neighbor-based techniques have found good application in the fields of clustering and outlier detection. For example, an effective hierarchical clustering algorithm named *CHAMELEON* [21] was designed based on a k-nearest neighbor graph whose nodes and edges represent objects and the k-nearest neighbor relations of these objects, respectively.

Multimedia Databases

With the availability of more and more online multimedia data, there has been a growing requirement for the development of multimedia database. In multimedia databases, the data are usually preprocessed and summarized as a set of features, which are then represented by a feature vector. The similarity between two objects is defined as the distance (e. g., Euclidean distance) between two feature vectors. For example, one similarity query may request the system to find the most similar image to a given image. The similarity query can be implemented by finding the objects with feature vectors that are closest to the feature vector of the query object. Therefore, the nearest neighbor search is one of the essential techniques in multimedia databases.

Document Retrieval

In the field of document retrieval, the major problem is to find the documents from the corpus that are relevant to the given query. This is in essence a nearest neighbor search problem. Currently, nearest neighbor techniques have already been successfully applied in this field. The popular vector model is a good example. In this model, each document is represented as a feature vector, and the query is also represented as a feature vector. The cosine similarity is defined as the distance function between two feature vectors. The retrieval algorithm searches for documents with feature vectors that are closest to the feature vector of the query.

Future Directions

With the development of spatial location detection techniques, there are more and more applications for *NN* queries. Recently, there have been two interesting and emergent directions: one is *NN* queries in road networks and the other is *NN* queries over moving objects.

Most of traditional *NN* algorithms are designed based on the Euclidean distance. In some real-world applications, however, data objects are constrained by spatial networks. In such cases, the actual distance between two objects corresponds to the road network distance. It is an interesting topic to efficiently compute *NN* queries in this distinct metric space.

In recent years, the prevalence of inexpensive mobile devices (e. g., *PDA* and mobile phones), together with the advances in sensor networks and location techniques, enables a location aware environment, where all objects of interest can determine their locations. Under such environments, the objects and query points are continuous changing their locations, and updates are periodically sent to the databases. It is a challenging task to design efficient and scalable algorithms to compute the *NN* queries over continuously moving objects.

Cross References

► Indexing, Hilbert R-tree, Spatial Indexing, Multimedia Indexing

References

1. Arya, S. Mount, D.M., Netanyahu, N.S., Silverman, R., and Wu, A.Y.: An optimal algorithm for approximate nearest neighbor in fixed dimensions. J. ACM **45**(6), 891–923 (1994)
2. Hu, H., and Lee, D.L.: Range nearest-neighbor query. IEEE Trans. on Knowl. Data Eng. **18**(1), 78–91 (2006)
3. Roussopoulos, N., Kelley, S., Vincent, F.: Nearest neighbor queries. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 71–79 (1995)
4. Fix E., Hodges, J.L.: Discriminatory analysis: non-parametric discrimination: consistency properties, Report No. 4. USAF School of Aviation Medicine, Randoph Field (1951)
5. Fix E., Hodges, J.L.: Discriminatory analysis: non-parametric discrimination: small sample performance, Report No. 11. USAF School of Aviation Medicine, Randolph Field (1952)
6. Lawler, E.L., Lenstra, J. K., Rinnooy Khan, A.H.G., Shmoys, D. B.: The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization. John Wiley & Sons, Chichester (1985)
7. Rosenkrantz, D.J., Stearns, R.E., Lewis II, P.M.: An analysis of several heuristics for the traveling salesman problem. SIAM J. Comput. **6**, 563–581 (1977)
8. Sproull, R.: Refinements to nearest neighbor searching in K-dimensional trees. Algorithmica **6**, 579–589 (1991)
9. Arya, S., Mount, D., Netanyahu, N., Silverman, R., Wu, A.: An optimal algorithm for approximate nearest neighbor searching. J. ACM **45**(6), 891–923 (1998)
10. Samet, H.: The Design & Analysis of Spatial Data Structures, Addison-Wesley, Reading, MA (1989)
11. Samet, H.: Applications of Spatial Data Structures, Computer Graphics, Image Processing and GIS, Addison-Wesley, Reading, MA (1990)
12. Hjaltason, G.R., Samet, H.: Distance browsing in spatial databases. ACM Transaction on Database. Syst. (TODS) **24**(2), 265–318 (1999)
13. Indyk, P., Motwani, R.: Approximate nearest neighbor: towards removing the curse of dimensionality. In: Proceedings of the Symposium on Theory of Computing, pp. 604–613 (1998)
14. Datar, M., Immorlica, N., Indyk, P., Mirrokni, V.: Locality sensitive hashing scheme based on p-stable distributions. In: Proceedings of the ACM Symposium on Computational Geometry, pp. 253–262 (2004)
15. Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In: Proceedings of the Symposium on Foundations of Computer Science, pp. 459–468 (2006)
16. Ooi, B.C., McDonell, K.J., Sacks-Davis, R.: Spatial kd-tree: An indexing mechanism for spatial databases. In: Proceedings of the IEEE COMPSAC Conference, Tokyo, pp. 433–438 (1987)
17. Guttman, A.: R-Trees: A dynamic index structure for spatial searching. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 47–57 (1984)
18. Finkel, R.A., Bentley, J.L.: Quad trees: A data structure for retrieval on composite keys. Acta Informatica **4**(1), 1–9 (1974)
19. Berchtold, S., Bohm, C., Keim, D.A., Kriegel, H.: A cost model for nearest neighbor search in high-dimensional data space. In: Proceedings of ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), Tucson, Arizona, pp. 78–86 (1997)
20. Cover, T.M., Hart, K.J.: Nearest Neighbor Pattern Classification. IEEE Transaction on Inf. Theory. **13**(1), 21–27 (Jan, 1967)
21. Karypis, G., Han, E.H., Kumar, V.: CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. IEEE Comput. **32**(8), 68–75, (1999)

Nearest Neighbor Query, Definition

FENG CHEN, CHANG-TIEN LU
Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA

Definition

Given a set P of n points in a d-dimensional space $\Re^d$ and a query point $q \in \Re^d$, a nearest neighbor (*NN*) query is to find the set of nearest points to the query point q. Formally, $NN\,(q) = \{p|p \in P, \forall o \in P, (o \neq p), |qp| < |qo|\}$. In most cases, $\Re^d$ refers to a d-dimensional Euclidean space.

Main Text

Nearest neighbor search is one of the oldest problems in computer science. The *NN* algorithm was one of the earliest algorithms used to determine a solution to the traveling salesman problem (*TSP*). The *NN* query first appeared in Geographic Information Systems around 1990 and has now already become a frequently encountered query type in this field. Recently, several NN algorithms have been proposed for *NN* queries based on spatial access methods (e. g., *quad*-trees, *R*-trees). For all these algorithms, depth-first search (*DFS*) and breadth-first search (*BFS*) algorithms are two most common branch-and-bound paradigms for processing *NN* queries.

With the development of spatial location detection techniques, there are more and more applications for *NN* queries. Because of the appearance of new environments, it is required to extend the traditional *NN* algorithms from the traditional environment to these new environments. Recently, there have been two interesting and emergent directions in the field: one is NN queries in road networks and the other is *NN* queries over moving objects.

Cross References

► Indexing, Hilbert R-tree, Spatial Indexing, Multimedia Indexing

References

1. Roussopoulos, N., Kelley, S., Vincent, F.: Nearest Neighbor Queries. In: Proc. ACM SIGMOD Int'l Conf. on Management of Data, pp. 71–79 (1995)

Nearest Neighbor Search

► Trip Planning Queries in Road Network Databases

Nearest Neighbors

► Constraint Databases and Data Interpolation

Nearest Neighbors Problem

D.J. O'Neil
Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA

Synonyms

Nearest neighbor; All-nearest-neighbors; All-k-nearest neighbors; Reverse-nearest-neighbor-problem; Reverse-k-nearest-neighbors aNN; akNN; rNN; rkNN; Range query; Spatial network; Balanced box decomposition tree (spatial index)

Definition

Given a set of **n** points and a query point, **q**, the nearest-neighbor (NN) problem is concerned with finding the point closest to the query point. Figure 1 shows an example of the nearest neighbor problem. On the left side is a set of **n** = 10 points in a two-dimensional space with a query point, **q**. The right shows the problem solution, **s**.

The nearest-neighbor problem also includes the following problems:

- **k-nearest-neighbors (kNN)**: Given a value **k** ⩽ **n**, kNN finds the k nearest objects to the query object. In most cases, the solution is the ordered k-nearest neighbors where the objects in the solution are ranked closest to farthest from the query point.
- **all-nearest-neighbors (aNN)**: aNN is essentially NN applied to every point in the dataset.
- **all-k-nearest-neigbors (akNN)**: akNN is kNN applied to every point in the dataset. Both akNN and aNN are usually used when NN queries will be applied to the data many times.
- **reverse-nearest-neighbor (rNN)**: given a query point, **q**, rNN finds all points in the dataset such that **q** is their nearest neighbor.
- **reverse-k-nearest-neighbor (rkNN)**: rkNN is similar to rNN except that it finds all points such that the query point, **q**, is in the set of their k-nearest-neighbors.

Within the GIS context, nearest neighbor problems are common in spatial queries where a user wants to know the closest objects, usually of some type, nearest to a query point. An example of this would be a query for finding what gas stations are near a particular address.

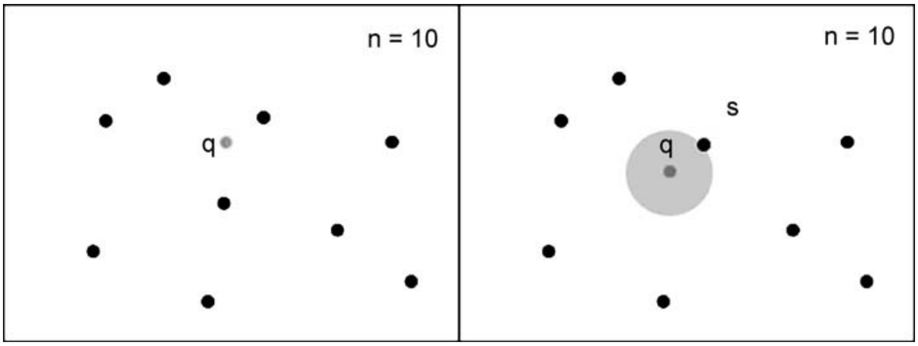
Historical Background

The nearest-neighbor heuristic was developed as a greedy approach to approximating the traveling salesman problem (TSP). This problem dates as far back as the 1800s when related problems in graph theory were discussed by the Irish mathematician, Sir William Rowan Hamilton, and by the British mathematician, Thomas Penyngton. The formal definition of the problem was first studied by Karl Menger at Harvard and Vienna in the 1930s. The TSP eventually became one of the classic graph problems in computer science [3]. More recently, the nearest neighbor problem has appeared in GIS, artificial intelligence, pattern recognition, clustering techniques, and outlier detection.

Scientific Fundamentals

The nearest-neighbor algorithm is as follows:

N



Nearest Neighbors Problem, Figure 1 An example of a nearest-neighbor problem domain and solution

Given:
- a set of **n** points.
- a query point, **q**.
- an initial minimum computed distance, **min_distance** $= \infty$.

For each point, n_i, i = 0, 1, ..., n−1:
1. Compute the distance between points **q** and $\mathbf{n_i}$.
2. If the computed distance, **d**, is less than **min_distance**, then set **min_distance = d** and **current_nearest_neighbor = ni**.

Solution: current_nearest_neighbor

The set of points can be defined in any space so long as a distance function is defined. Common distance functions in metric spaces are the Euclidean and Manhattan distances. The Euclidean distance is the distance found by determining the length of the straight line segment that joins the points. The Manhattan distance is the shortest summed distance between the points using segments that are right angles to each other. The relationship between these two distance functions is that they represent the two paths between the points using a right triangle where the Euclidean distance is the hypotenuse and the Manhattan distance represents the legs.

The nearest-neighbor algorithm has two classical contexts. The first has to do with simply finding the nearest neighbor of some query point and the second uses neighbors as a simple classification technique.

Consider an example of the first type, such as finding the nearest gas station. Figure 2 shows a search from Google Maps for gas stations near the Metrodome in Minneapolis, MN. Here the **n** points consist of every geographical object that has an address. The query point is the Metrodome. In GIS applications such as this, the data points are usually filtered to reduce the number of candidate points. This might include only considering points within a certain geographic region (e. g. Minneapolis, MN) and of a specific service type (e. g. Gas Station). The nearest search results are the nodes labeled **A**, **B**, **C**, and **D** with **A** being the closest, then **B**, and so on. This problem is slightly different than the simple example illustrated in Fig. 1. Because gas stations can only be reached by driving on the road, one cannot simply compute the Euclidean distance between the start point and candidate gas stations. In this example the distance function needs to consider multiple (reasonable) paths, constraints such as one-way streets, speed limits, and possibly other factors.

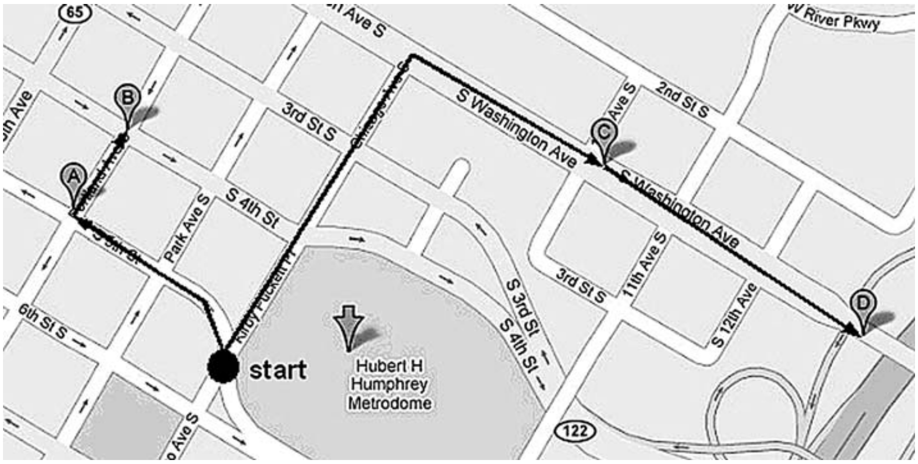
This example also demonstrates the k-nearest-nearest neighbors search. In this case, **n** could be all the gas stations in the world and **k** could be four. The solution to the problem is the gas stations identified by nodes **A**, **B**, **C**, and **D**.

Another common use of the nearest-neighbor problem is simple classification of a d-dimensional dataset. For example, suppose one wanted to predict which way someone would vote based on the following ($\mathbf{d} = 5$):
- Age
- Height
- Weight
- Years of education
- Income level

Furthermore, suppose a populated sample was collected with this data along with political affiliation; for simplicity, assume they can only be a Democrat or a Republican. Then, given the above data for an unclassified sample, the nearest-neighbor algorithm would use a distance function, f, to determine which 5-tuple in the classified dataset set is closest. The predicted classification of this new data point would be the same as the classification of its nearest neighbor.

A major problem with the simple nearest-neighbor algorithm is that it considers the entire set of **n** points for every execution. However, consider the Ann and Aknn problems where the same dataset is used **n** times. It would be beneficial to get some information about the dataset and the relationship between points in order to decrease the search space.

One of the simpler methods is to use Voronoi Diagrams. Given a set of points, a Voronoi Diagram can be constructed such that every point in the region surrounding a data point is closest to that data point than any other. Figure 3



Nearest Neighbors Problem, Figure 2 Finding the nearest gas station. (image from Google Maps. ©2006 Google – Map data ©2006 NAVTEQ™)

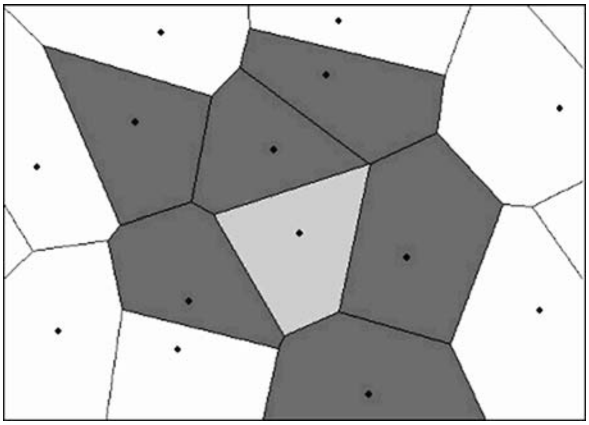
depicts a Voronoi Diagram. Consider the query point, **q**. To reduce the search space, one needs only to consider those points whose representative regions (dark gray) border the region of the query point (light gray). For values of **k** greater than the number of bordering regions, the algorithm could search a subset of regions that border the union of the original candidate regions, and so on.

Key Applications

The nearest-neighbor problem and its variants are used in a variety of application domains.

Spatial Network/Geographical Information Systems (GIS)

A spatial network is any network of spatial objects. GIS is an application of spatial networks that consist of points of various types of geographical data such as cities, parks, gas stations, etc … Spatial network problems, such as GIS, are different than classical Euclidean space queries because they need to consider spatial network distance. Spatial distance might be more concerned with time, whereas geometric distance exclusively considers Euclidean distance. Consider the nearest gas stations problem described earlier: in addition to length traveled using roads, GIS time distances might also consider speed limits and traffic jams depending on the time of day. Because of these added metrics, the computed distance may not be the shortest [9].
Due to increased storage space and processing power, spatial databases have to continued to grow in size and complexity. GIS applications, such as Google's Google Earth, contain detailed images and maps, extensive information on businesses and business profiles, and a huge address database. Because of the sheer volume of data to examine for nearest neighbor queries, new approaches to solving nearest neighbors have been proposed. These include approximation techniques that use reasonable assumptions to decrease the search space and novel data structures (k-d Trees, Range trees) designed with nearest-neighbors in mind.



Nearest Neighbors Problem, Figure 3 An example of a Voronoi Diagram

Artificial Intelligence (AI)/Pattern Recognition

Pattern recognition is similar to the classification method discussed earlier. In AI, pattern recognition has been applied to such problems as learning and vision using example-based methods such as the nearest-neighbor. For machine learning, assuming the system has a large dataset of classified tuples, given a new data tuple, the nearest-neighbor algorithm can be applied on the classified data set. The new tuple will take on the classification of the nearest classified tuple. This method has been extended to use knn where the k-nearest tuples are located and the new tuple takes on the dominating classification of the k nearest tuples (ties broken by increasing or decreasing **k**).

The same method can be applied to images. For example, in [10], a hand pose was estimated using 100,000 hand images with known pose positions using the Chamfer distance.

Clustering

The clustering problem is concerned with identifying associations between data points. Using the nearest-neighbors algorithm and a maximum distance threshold, the nearest neighbors can be incrementally identified from closest to farthest until the threshold has been reached. Clustering is used in many disciplines, ranging from marketing to geography to census data. [11]

Outlier Detection

An outlier is a deviating point that falls outside the overall pattern of a data set. In clustering and statistical problems, outliers are considered noisy data and are often removed. However, in data mining, an outlier may indicate an important anomaly. This has been used in intrusion detection systems where a normal pattern of facility use is established and outliers are classified as possible intruders. AkNN has been used in this problem by first identifying clusters in a controlled dataset. Then, given a new data point, **p**, the knn are computed. If computed kNN falls into more than **T** clusters, then the data point is considered to be an outlier. [12]

Statistics

Because nearest neighbors can provide a quick and intuitive method to classify new data, the method has been applied to statistical methods. A primitive approach might simply assign the new point the classification of its classified NN. Using kNN, confidence intervals and hypothesis tests for classifications can be applied to new data points by finding the set of **k** classified points nearest to the new data point.

Business

In addition to some of the applications listed above, such as AI and statistics, nearest neighbor approaches have been applied to many common business problems. For example, suppose a business would like to open up a new branch at one of several locations. A rkNN search could be applied to the candidate locations to predict which would have the largest customer base using the demographics of its surrounding geographic locations. This search can also be extended to predict how many service requests a particular business location might experience.

Future Directions

As a brute force technique, the traditional nearest-neighbor algorithm is not very useful with large datasets, especially if the dataset is going to be reused. When considering how much information is contained in many GIS applications, it becomes apparent that searching the entire search space for a objects near the query point would be extremely costly and ineffective. Current research in nearest-neighbor algorithms is concerned with determining ways to approximate the solution and using nearest-neighbors as an approximation for other problems. For example, in [8], highly dimensional data points are first restructured into a balanced box-decomposition (BBD) tree where points are classified by hyper-rectangles with a bounded width vs. height ratio. Similar to an R-tree, the BBD tree can be used to quickly decrease the search space. Knn has also been used in GIS applications to estimate the most likely candidates of a spatial query. Finding the knn using Euclidean distance is straightforward and simple and saves more time than computing knn using complex spatial distances.

There has also been research in the appropriateness of applying Euclidean and/or Manhattan distance functions in highly dimensional spaces. As noted in [14], when dimensionality increases (10–15 dimensions), the distance to the nearest and farthest data points approach each other using traditional distance functions. Highly dimensional data also complicates the use of traditional spatial data structures. The authors in [13] observed that as dimensions increase, the overlap of minimum bounding boxes in R-trees increases significantly. They propose a new data structure, called the X-tree, that aims to keep data as hierarchical as possible to reduce overlap using the concept of supernodes.

Cross References

► Voronoi Diagram

Recommended Reading

1. Levitin, A.: Introduction to The Design and Analysis of Algorithms. Addison Wesley (2003)
2. The Algorithm Design Manual. Nearest Neighbor Search. Jun. 2 (1997)
3. Traveling Salesman Problem. History of the TSP. Jan. (2005)
4. Google. Google Maps. October 12, (2006). Search on "gas stations" near the "metrodome"
5. Chavez, E., Figueroa, K., et al.: A Fast Algorithm for the All k Nearest Neighbors Problem in General Metric Spaces
6. Abhinandan, D., Gehrke, J., et al.: Approximation techniques for spatial data. Proceedings of the 2004 ACM SIGMOD international conference on Management of data. Paris, France. SESSION: Research sessions: spatial data. Pgs. 695–706. (2004)

7. Tunkelang, D.: Making the Nearest Neighbor Meaningful. SIAM WOrkship on Clustering High Dimensional Data and its Applications. (2002)
8. Arya, S.: An optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions. Journal of the ACM (JACM) (1998)
9. Ku, W.-S., Haojun, W., et al.: ANNATO: Adaptive Nearest Neighbor Queries in Travel Time Networks. 7th International Conference on Mobile Data Management 2006. pp. 50. May 10–12 (2006)
10. Athitos, V.: Approximate Nearest Neighbor Retrievel Using Euclidean Embeddings. Approximate Nearest Neighbors Methods for Learning and Vision. Dec 13 (2003)
11. Painho, M., Bacao, F.: Using Genetic Algorithms in Clustering Problems. GeoComputation 2000 (2000)
12. Hautamaki, V., Karkkainen, I., et al.: Outlier Detection Using k-Nearest Neighbor Graph. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004. vol. 3. Aug. 23–26, pp. 430–433, (2004)
13. Berchtold, S., Keim, D., A., Kriegel, H.-P.: The X-tree: An Index Structure for High-Dimensional Data. In: Proceedings of the 22nd International Conference on Very Large Databases (1996)
14. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is 'Nearest Neighbor' Meaningful? Lecture Notes in Computer Science (1999)

Negatively-Correlated

► Patterns, Complex

Negotiation

► Participatory Planning and GIS

Neighborhood

► Homeland Security and Spatial Data Mining

Neighborhood Segmentation

► Geodemographic Segmentation

Nested-Loop, Blocked

► Skyline Queries

.NET Framework

► MapWindow GIS

Network Data Model

► Road Network Data Model

Network GIS Performance

PHIL YANG[1,3], JOHN EVANS[2,3]
[1] Joint Center for Intelligent Spatial Computing College of Science, George Mason University, Fairfax, VA, USA
[2] Global Science & Technologie Inc., Greenbelt, MD, USA
[3] NASA Geosciences Interoperability Office, Goddard Space Flight Center, Greenbelt, MD, USA

Synonyms

Computing performance; Quality of services

Definition

Network geographic/geospatial information systems (GIS) performance (NGP) refers to the level of quality of services (QoS) of a network GIS. NGP includes both the efficient use of network-GIS resources (CPU, memory, massive storage, geospatial data and geospatial services) and the perception of speed of a network GIS [1]. Network GIS is a GIS where the geospatial data and geospatial processing are distributed across a computer network [2]. The performance [3] can be broken down into different factors, such as user-perceived system response time, system reliability, system extensibility, and system service quota. Different practitioners with different roles for a network GIS have different views of NGP (Table 1).

All these perspectives are somehow dependent on the different user's perceived performance; for example, how much time should be spent on waiting for a task to be accomplished (user)? Could a system support prioritizing different applications according to different types of users (designer or developer)? Is it possible to guarantee a certain level of performance if a critical application needed to be supported, for example, police-car routing or battlefield support (host/owner)?

An experience study in 2001 on user's waiting time for web browser response discovered an 8-s rule: the amount of time for a general user to wait for a web response without giving up is about 8 s [4]. Nevertheless, different users will have different requirements.

Performance Bottlenecks of a Network GIS

The bottlenecks of NGP include: (1) availability of geospatial data or services [5], (2) server capabilities to respond

Network GIS Performance, Table 1 Different network-geographic/geospatial information systems (GIS) practitioners' perspectives of network-GIS performance (NGP)

Roles	Different Perspectives	Mainly Rely on	Restricted by
Owner	How much to pay for serving a certain number of users?	The designer and developer to provide solution, and the end-users to provide feedback	Available budget, resources to support the procurement of a network GIS
Designer	What strategy/architecture should be chose to meet the owner's NGP requirements?	Owner requirements and designer's experience	Available technologies, architecture, COTS
Developer	How to implement a reliable and stable system with designed functions?	Designer's design and developer's experience	Available developing tools and developer's knowledge
User	How much time is needed to wait for a system response?	System experienced and the expectation	Available GUIs and users' experiences
Host	How much resources are needed to provide needed service?	Infrastructure and the design requirements	Outgoing network bandwidth, server capacity

COTS Commercial off the shelf, *GUI* graphical user interface

to massive numbers of users' simultaneous accesses, (3) network bandwidth when big volumes of data need to be transmitted, and (4) software efficiency in processing geospatial functions. The first bottleneck is being researched as interoperability and supported by purchasing more data and services. The second bottleneck is tackled by utilizing supercomputers or massive cluster computing [6]. The third is addressed by improving the bandwidth of the computer network, for example, the former kilo-bps (bits per second) phone connection is upgraded to mega-bps for home connection and the mega-bps backbone is upgraded to over gega-bps and tera-bps. The fourth bottleneck is being solved by applying different techniques in developing and configuring geospatial software to improve the NGP.

Software Techniques for Improving Performance

Software techniques can be used to tackle all four bottlenecks, especially the fourth (Table 2).

Mechanisms for Ensuring Performance

Different types of performance are also needed. For example, consider a routing system that integrates different user requirements based on a developed spatial data infrastructure (SDI): (1) travelers may need responses within minutes or hours of when they plan to drive, (2) postmen may need responses within minutes before driving to the next stop every time a package is delivered, (3) policemen may need responses within seconds to drive to a crime scene. The best case in this scenario would be that everyone get their responses within their allowable time limit. In this scenario, how to guarantee the policeman get a real-time response needs a performance-ensuring mechanism. To ensure NGP for urgent GIS applications, software techniques on scheduling could be applied to reserve resources or make sure high priority is given to urgent tasks. The resources QoS-supporting protocols for computer networks can be applied here to ensure NGP. These protocols maintain a resource loading status and incoming requirements will be assigned a priority. Therefore, tasks with higher priority can be chosen for processing when geospatial resources are available. Some popular network protocols can be adopted to ensure NGP (Table 3).

Historical Background

Performance has been used as a mechanism to measure the time needed to accomplish a task since the inception of the computer in 1941, and for scheduling tasks input to the computer [9]. Since then, performance has been widely used as a criterion for procuring an information system, and in designing and developing such a system [10]. With the popularity of GIS and network GIS, and the construction of the National and Global Spatial Data Infrastructures (NSDI, GSDI), performance becomes an important factor in measuring the success of a network GIS. For example, the Federal Geographic Data Committee (FGDC) Clearinghouse has a web interface to check the status of FGDC clearinghouse nodes [11]. Performance is also used in differentiating the geospatial information services provided by different network GIS applications [12].

Scientific Fundamentals

The scientific fundamentals of NGP include support from at least four different aspects: GIS, computer network research, high performance computing, and cartography principles.

Network GIS Performance, Table 2 Methods applicable to improve NGP (adopted from [7,8])

Techniques	Bottlenecks addressed		Overhead
Pyramid, Cut, Hash index	Software efficiency on large image management	Data management server	Maintain metadata
Cache	Network bandwidth, data I/O	GUI and GIS server	Memory usage
Dynamic Request	User perceived response time	Client	Smart client
Multithreading	Software efficiency	GUI and GIS server	Threads scheduling
Cluster/Grid Computing	Server computing capacity	GIS server	User session management
Compression	Network bandwidth	GIS server and client	Compression, decompression
Interoperability	Availability of data and services	GIS server and client	Reformatting data or reprogram system interface

Network GIS Performance, Table 3 Networking protocol for ensuring the NGP [3]

Methods	Working Mechanism	Advantage	Disadvantage
Best effort	Requests are attempted to be served at the best capacity of the system	Efficient system usage	No ensured performance
FCFS	First come first serve	Based on the waiting philosophy	No priority added to specific tasks
Integrated Service	Resources are first reserved; Remaining resources can be allocated by FCFS	Have priority and guaranteed performance	System resources may be wasted if reserved resources are not used
Differentiated Service	Requests are marked at different priorities. Resources will be allocated to the priorities first, and then could be FCFS	Best used of the resources	Low priority requests may never be responded if there are too many high priority requests

FCFS First come first serve

Performance is reflected by a network GIS application, which requires the performance of a certain function (for example, routing a driving route from Washington DC to San Francisco) within an acceptable time (for example, under 8 s) and returning accurate results (a correct routing map with the necessary details). The GIS applications also provide testbeds for benchmarking the performance of a network GIS.

Research on computer networks has lots of similarities with research on network GIS. The methods and algorithms used for computer network routing, scheduling, and assuring can also be used to ensure the performance of a network GIS.

High performance computing provides the necessary computing power to support a time-consuming network GIS. For example, the Google Earth online mapping application is supported by thousands to millions of computers (or CPUs). An earth science simulation that would run for a week on a desktop computer could be finished within minutes on a high-performance computer.

Cartography principles provide the fundamental methods in separating computing objects into pieces to be calculated on different computers and abstracting computing objects into different levels to facilitate the needs of different users from the global to the regional level, for example, visualizing the globe on a desktop and visualizing the National Mall on a desktop from the same data source.

Key Applications

NGP can be used in many application domains, most notably the infrastructure and emergency management communities.

Spatial Data Infrastructure

NSDI and GSDI have been researched and developed for over 10 years to share geospatial information resources through computer networks. When massive numbers of users are accessing the infrastructure, NGP will be an important mechanism to measure the success of a GSDI/NSDI and provide reliable services to urgent needs.

Decision Support Systems

NGP can be used in decision support systems to schedule the information integration and user access where a variety of geospatial information are integrated and a variety of user needs are supported.

Daily Life

NGP can be used to help daily life by prioritizing tasks in geospatial information processing, such as travel places, meeting locations, and other complex geospatial information needs.

Defense

NGP can be used to help schedule the various defense needs of global geospatial information networks [13] to be constructed by the national spatial intelligence agencies for control, planning and battlefield information.

Emergency Management

NGP can be used to differentiate GIS services provided to different personnel involved in emergency management, such as firefighters, doctors, and the police.

Geospatial Computing Infrastructure

NGP can help to prioritize users' access to geospatial computing infrastructure if it is accessed by simultaneous users for services. For example, real-time coastal management forecasting, weather forecasting, emergency management, and disaster management parameter calculations.

Popular Geospatial Services

NGP can be used for those popular services, Google Earth and Virtual Earth [14], for example, to provide reliable services to paying customers and emergency needs, while still providing acceptable services to the general public.

National and Global Geospatial Applications

NGP can be used to provide reliable services to different users, communities, countries, and regions to strategize the usage and maximize the benefit of such national and global geospatial applications.

Future Directions

NGP is an emerging concept with the popularization of geospatial information services, interoperability, spatial information infrastructure, and geospatial computing infrastructure. It provides a promising mechanism for utilizing limited resources to support unlimited and differentiated requirements according to the nature of the requirement, the submitting user, regions, and user community. However, to put it into practical use, more research, testing, installment, evaluation, and benchmarking have to be conducted.

The evolution of NGP will rely on future work in (1) the architectural evolution of GIS and performance research, (2) the sharing strategies of supercomputing and other computing research [15], (3) user cognitive research, and (4) the evolution of priority scheduling in the computing sciences.

Cross References

► Distributed Geospatial Computing (DGC)
► Internet-Based Spatial Information Retrieval
► Spatial Data, Indexing Techniques

Recommended Reading

1. Apple Developer Connection: What Is Performance? (2006) http://developer.apple.com/documentation/Performance/Conceptual/PerformanceOverview/DevelopingForPerf/chapter_2_section_2.html. Accessed 29 Aug 2007
2. Yang, C., Wong, D., Kafatos, M., Yang, R.: Network GIS. In: Qu, J., Gao, Y., Kafatos, M., Murphy, R., Solomonson, V. (eds.) Earth Science Satellite Remote Sensing, vol. 2, pp. 252–269 Springer and Tsinghua University Press, Beijing, China (2006)
3. Evans, J., Clarence, F.: Deploying IP and MPLS QoS for Multiservice Networks: Theory and Practice. Morgan Kaufmann, San Francisco, CA (2007)
4. Submit Corner (2001) 8 Second Rule. http://www.submitcorner.com/Guide/Bandwidth/001.shtml. Accessed 29 Aug 2007
5. Yang, C., Tao, V.: Distributed geospatial information service. In: Rana, S., Sharma, J. (eds.) Frontiers of Geographic Information Technology, chap. 5, pp. 113–130. Springer, London (2005)
6. Healey, R., Dowers, S., Gittings, B., Mineter, M.: Parallel Processing Algorithms for GIS. Taylor Francis, Bristol, PA (1998)
7. Yang, C., Wong, D., Kafatos, M., Yang, R.: Implementing computing techniques to accelerate network GIS. In: Li, D., Xia, L. (eds.) Proceedings of the SPIE Vol. 6418, 64181C, Geoinformatics (2006): GNSS and Integrated Geospatial Applications, Wuhan, China, 27–29 Oct 2006
8. Yang, C., Wong, D., Yang, R., Kafatos, M., Li, Q.: WebGIS performance improving techniques. Int. J. Geogr. Inf. Sci. **19**, 319–342 (2005)
9. Kempf, K.: Historical Monograph: Electronic Computers Within the Ordnance Corps (1961) Available via: http://ed-thelen.org/comp-hist/U-S-Ord-61.html. Accessed 29 Aug 2007
10. Barber, S. Beyond Performance Testing Part 2: A Performance Engineering Strategy (2004) http://www-128.ibm.com/developerworks/rational/library/4215.html. Accessed 29 Aug 2007
11. FGDC Clearinghouse: Node performance measurement (1999) http://www.fgdc.gov/clearinghouse. Accessed 20 Jul 2005
12. Yang C., Evans J., Bambacus M., Cole M., Alameh N., Marley S.: Spatial Web Portal: Architecture and Example. PE&RS 73(6):691–698 (2006)
13. Fuller, G.W.: Capabilities for a Future National Intelligence Enterprise (2005)http://www.garywfuller.net/pubs/capabilities_nie.html. Accessed 29 Aug 2007
14. Tao, V.: Microsoft Virtual Earth: New Horizons in on-line mapping. Keynote presented at GeoInformatics'2006, Wuhan, China, October 28–29, 2006
15. Aydin, G., Sayar, A., Gadgil, H., Aktas, M.S., Fox, G.C., Ko, S., Bulut, H., Pierce, M.E.: Building and applying geographical information system grids. In: Proceedings of GGF15 Workshop, Boston, MA, 3–6 Oct 2005

Network Voronoi

► Voronoi Diagrams for Query Processing

Networks, Spatial

► Contraflow in Transportation Network

Nine-Intersection Model

► Dimensionally Extended Nine-Intersection Model (DE-9IM)

NIST

► Spatial Data Transfer Standard (SDTS)

Non-Photorealistic Computer Graphics

► Visualization, Photorealistic and Non-photorealistic

Non-Photorealistic Rendering

► Visualization, Photorealistic and Non-photorealistic

Non-Raster Data Compression

► Data Compression for Network GIS

Nonresidential Population

► Population Distribution During the Day

Nonseparable Multivariate Models

► Hurricane Wind Fields, Multivariate Modeling

Non-Specificity

► Uncertainty, Semantic

NPR

► Visualization, Photorealistic and Non-photorealistic

NSDI

► National Spatial Data Infrastructure

Object Model

► Application Schema

Object Recognition

► Image Mining, Spatial

Object Reconstruction

► Photogrammetric Methods

Object Schema

► Application Schema

Object-Oriented

► Smallworld Software Suite
► Uncertain Environmental Variables in GIS

Object-Relational

► PostGIS

Objects with Broad Boundaries

ELISEO CLEMENTINI
Department of Electrical and Information Engineering, University of L'Aquila, L'Aquila, Italy

Synonyms

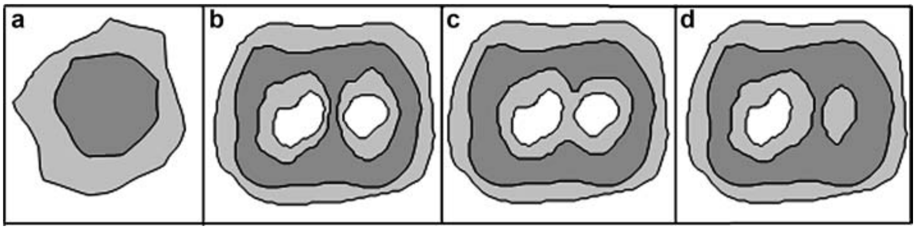
Spatial objects; Spatial data types with indeterminate boundaries; Vague boundaries; Uncertain boundaries; 3-value indeterminacy; Fuzzy sets; Probability theory; Egg-yolk model

Definition

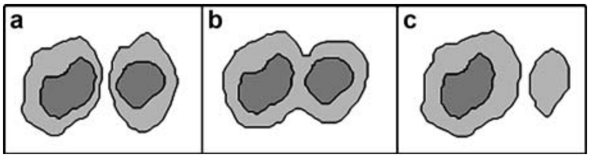
Objects with broad boundaries are spatial objects, whose crisp boundaries are replaced by an area expressing the boundary's uncertainty. There are two main interpretations for broad boundaries: (1) for positional uncertainty, the broad boundary represents the set of all possible positions among which the unknown boundary position is hidden; (2) for "fuzzy" boundaries, that is, boundaries that are by nature not crisp, the broad boundary represents their minimum and maximum extent. The main motivation for objects with broad boundaries is to record information about uncertainty together with the data. In this way, they represent a new geometric model that overcomes the limits of current spatial database models, which are a collection of lines (points, polylines and polygons). The geometric model of objects with broad boundaries takes into account a 3-valued indeterminacy of location (false, maybe, true), where "false" means that the point is not in the location, "true" means that the point is in the location and "maybe" means either that the point is with some probability in the location or that the point belongs, up to a certain membership value, to the location. Regions, lines and points with broad boundaries can be distinguished. In the case of lines, the broad interior can also be defined: a line with a broad boundary and broad interior is called an uncertain line.

Historical Background

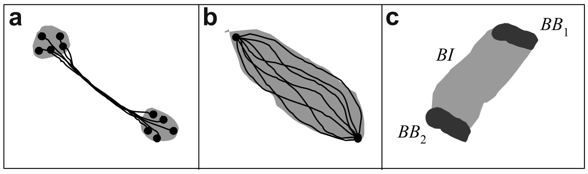
To represent uncertainty in spatial data, many models suggested the introduction of broad boundaries replacing crisp ones in the case of regions [2,5,7,10,14] and lines [1]. The advantage of objects with broad boundaries is that they can be implemented on existing database systems at reasonable cost, being a direct extension of existing geometric models. Objects with broad boundaries corresponding to a 3-valued indeterminacy of a region's location do not support any further hypothesis on the internal structure of the broad boundary. Other approaches utilize fuzzy sets [15] or probability theory [16], which describe the internal structure of the uncertain geometry with membership functions or probability distributions, respectively. These other



Objects with Broad Boundaries, Figure 1 Regions with a broad boundary



Objects with Broad Boundaries, Figure 2 Composite regions with broad boundaries



Objects with Broad Boundaries, Figure 3 Broad boundary (**a**), broad interior (**b**), and an uncertain line (**c**)

approaches require much effort to store data and also the operations become computationally expensive.

Research on objects with broad boundaries has mainly been committed to define models for topological relations [1,4,5], which are an extension of the following traditional models for describing topological relations between objects with crisp boundaries: the 9-intersection [9] and the calculus-based method (CBM) [6].

Scientific Fundamentals

A region with a broad boundary is an extension of a region with a crisp boundary (for the definition of the latter, refer to simple regions with holes [3]). The following definitions are based on point-set topology:

Definition 1. A *region with broad boundary* A is made up of two regions A_1 and A_2, with $A_1 \subseteq A_2$, where ∂A_1 is the *inner boundary* of A and ∂A_2 is the *outer boundary* of A.

Definition 2. The *broad boundary* ΔA of a region with broad boundary A is the closed subset comprised between the inner boundary and the outer boundary of A, i. e., $\Delta A = \overline{A_2 - A_1}$, or equivalently $\Delta A = A_2 - A_1^\circ$.

Definition 3. Interior, closure, and exterior of a region with broad boundary A are defined as $A^\circ = A_2 - \Delta A$, $\overline{A} = A^\circ \cup \Delta A$, $A^- = \mathrm{IR}^2 - \bar{A}$, respectively.

Examples of regions with broad boundaries are illustrated in Fig. 1. If simple regions are substituted by composite regions, the most general case of composite regions with broad boundaries is obtained (Fig. 2).

In the case of lines, it is necessary to take into account the uncertainty related to the endpoints of the line (broad boundary, Fig. 3a) and the uncertainty affecting the interior of the line (broad interior, Fig. 3b). These two independent aspects define the concept of uncertain line (Fig. 3c). They are defined as follows:

Definition 4. Given a simple line L, whose exact position in IR^2 is unknown but can be delimited by a bounded point set, the *broad boundary* of L is the union of two point sets $BB_1(L)$ and $BB_2(L)$ that contain the two endpoints of L.

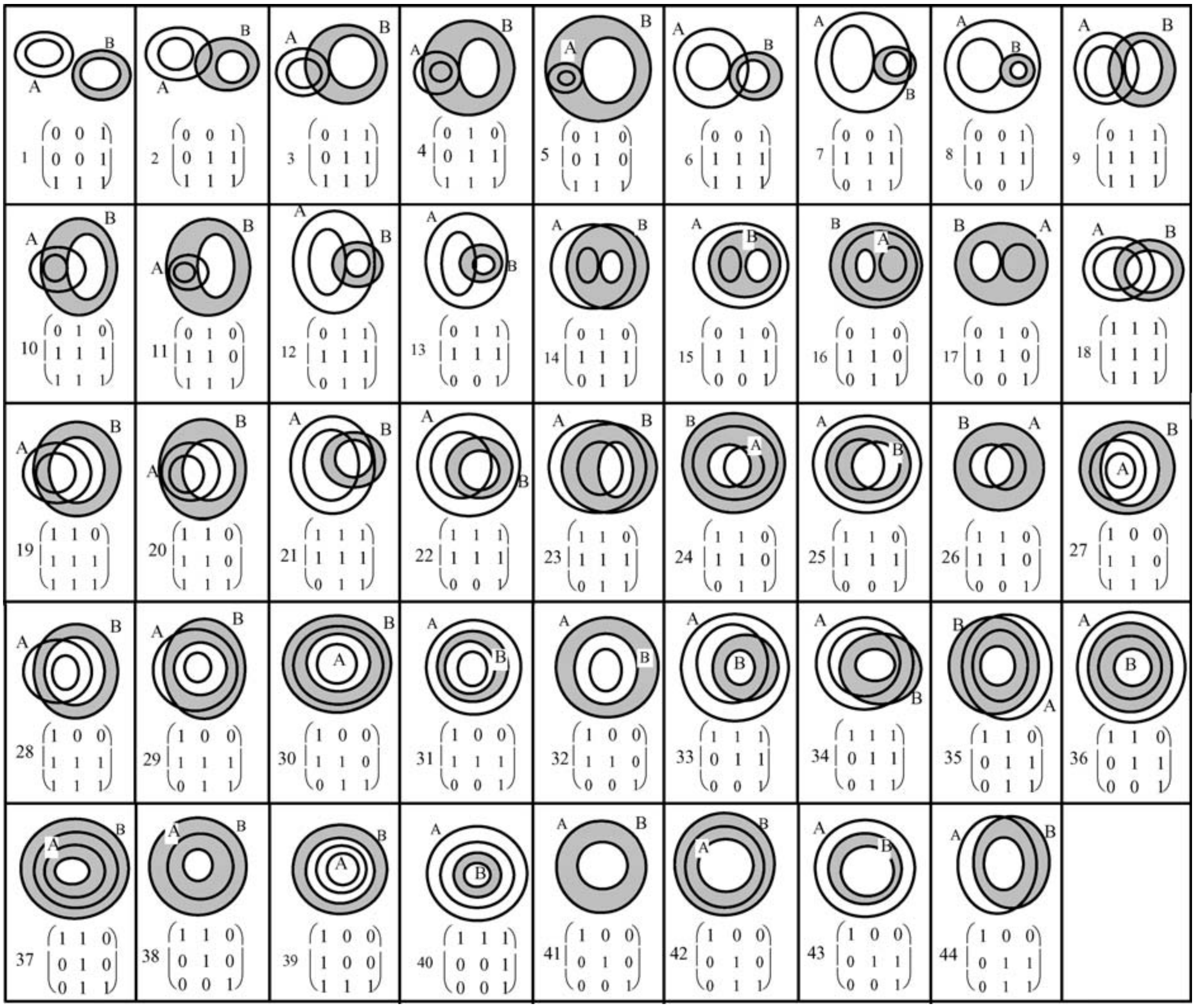
Definition 5. Given a simple line L, whose exact position in is unknown but can be delimited by a bounded point set, the *broad interior* of L is a point set $BI(L)$ that contains the interior of L.

Definition 6. Given a simple line L', an *uncertain line* L is made up of a broad boundary $BB_1(L')$ and $BB_2(L')$ and a broad interior $BI(L')$. The *boundary* of L, indicated with ΔL, is the union of $BB_1(L')$ and $BB_2(L')$. The *interior* of an uncertain line L, indicated by L°, is given by the difference $L^\circ = L - \Delta L$.

For a point-like spatial phenomenon, the main source of uncertainty is related to the indetermination of its position in space:

Definition 7. Given a point P', whose exact position in is unknown but can be delimited by a bounded point set, un *uncertain point* is a point set P that contains P'. The boundary of an uncertain point is empty, while the interior of an uncertain point is equal to the uncertain point.

The models for expressing the topological relations between spatial objects with uncertainty are defined starting from the main models that apply to crisp objects, namely, the 9-intersection [9] and the CBM [6].



Objects with Broad Boundaries, Figure 4 The 44 topological relations between simple regions with a broad boundary

The 9-intersection model applied to objects with a broad boundary is expressed by the following matrix:

$$M = \begin{pmatrix} A^\circ \cap B^\circ & A^\circ \cap \Delta B & A^\circ \cap B^- \\ \Delta A \cap B^\circ & \Delta A \cap \Delta B & \Delta A \cap B^- \\ A^- \cap B^\circ & A^- \cap \Delta B & A^- \cap B^- \end{pmatrix} .$$

An illustration of the 44 topological relations between regions with a broad boundary is given in Fig. 4 and the additional 14 relations for composite regions with a broad boundary in Fig. 5.

The topological relations of the CBM for objects with a broad boundary are defined as follows:

Definition 8. The relation *touch*(A, B) is true, if and only if:

$$(A^\circ \cap B^\circ = \emptyset) \wedge (A \cap B \neq \emptyset)$$

Definition 9. The relation *in*(A, B) is true if and only if $(A^\circ \cap B^\circ \neq \emptyset) \wedge (A \cap B = A)$.

Definition 10. The relation *cross*(A, B) is true if and only if:

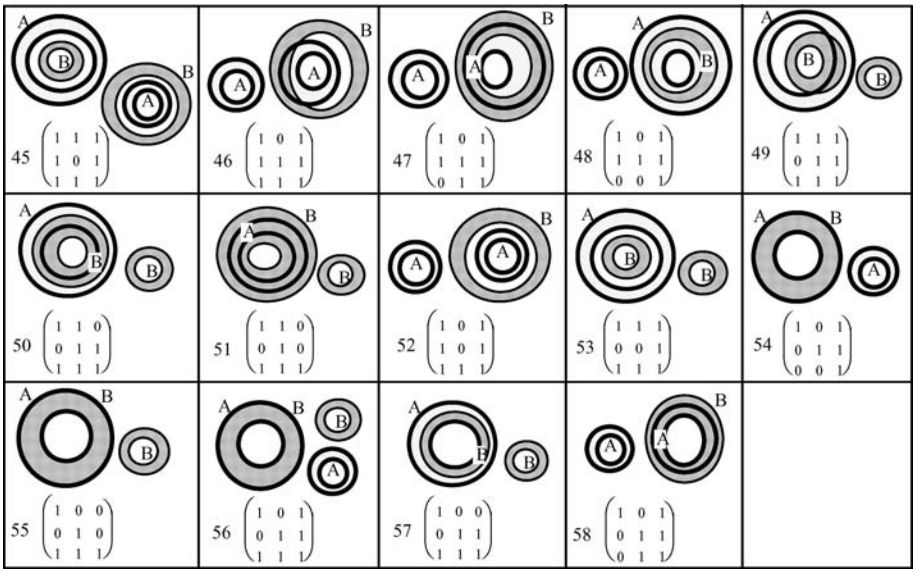
$$(\dim(A^\circ \cap B^\circ) < \max(\dim(A^\circ), \dim(B^\circ))) \wedge (A \cap B \neq A) \wedge (A \cap B \neq B) .$$

Definition 11. The relation *overlap*(A, B) is true if and only if:

$$(\dim(A^\circ) = \dim(B^\circ) = \dim(A^\circ \cap B^\circ)) \wedge (A \cap B \neq A) \wedge (A \cap B \neq B) .$$

Definition 12. The relation *disjoint*(A, B) is true if and only if $A \cap B = \emptyset$.

0



Objects with Broad Boundaries, Figure 5 The 14 additional topological relations for composite regions with a broad boundary

Spatial relations of the 9-intersection can be organized in a graph having a node for each relation and an arc for each pair of matrices at minimum topological distance [8], also called the *conceptual neighborhood* in [12]. The correspondence between the CBM relations and the 58 relations of the 9-intersection can be shown over the graph (Fig. 6) with a partition into four clusters.

The number of possible topological relations between uncertain lines is 146. In Fig. 7, they are listed by using a linear notation by rows for the values of the 9-intersection matrix.

Key Applications

A representation of a spatial phenomenon as an object with a broad boundary can have several semantic interpretations [17] and can be therefore used in a variety of applications.

Incomplete Representation of a Feature

In vector databases, if there are missing sides of a polygon or a line, due to omissions in digitization or imperfect data conversion, broad boundaries can be introduced to represent the possible location of the missing segments.

Conflicting Representations of a Feature

In the case of cadastral data, or representations of political boundaries (in general, the case of existing data that provide approximations to some real objects), broad boundaries can result from the merging of different representation of the same region.

Changing Representations of a Dynamic Spatial Phenomenon

If there are different observations of the same geographic phenomenon taken at different times, the core region may be interpreted as being the collection of locations that are in the region at all times, while the broad boundary may be interpreted as the collection of locations that are in the region at some, but not all, times.

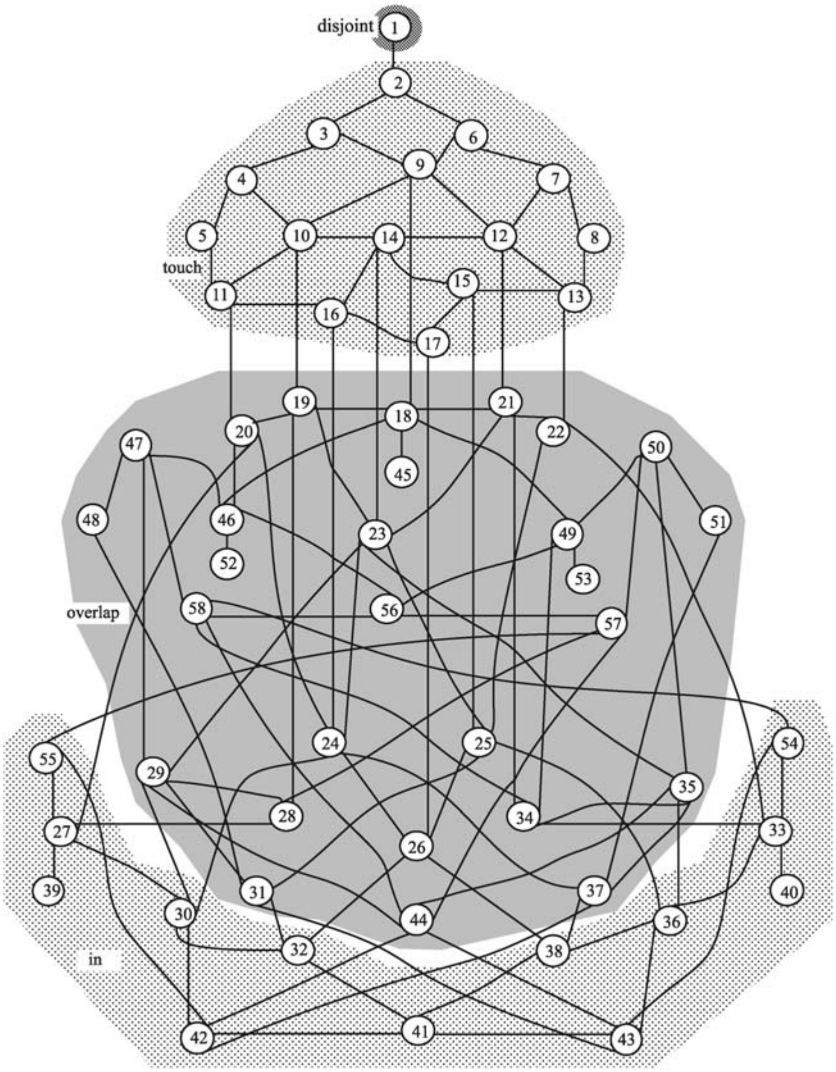
Imprecise Observation of a Spatial Phenomenon

At a certain resolution (i. e., a partition of the space in which locations indiscernible from the observation are grouped into the same elementary unit of the space), the elementary units impose a granularity of the underlying space. A region with a broad boundary can represent an imprecise observation of a region at a certain resolution.

Inherently Vague Representations of Real World Spatial Objects

Most geographical objects that are not manmade artifacts or conventions fall in this category. They may be represented by fuzzy sets, if a reasonable method of quantifying the membership function for the fuzzy set is available. A region with a broad boundary is interpreted as an approximation in which the broad boundary of the region represents the part of the object where the membership function gradually decreases from 1 to 0. There are two special cases:

Resulting from Scale Change These are spatial objects that only exist at coarser resolutions (small scale), but are



Objects with Broad Boundaries, Figure 6 The topological relation graph and the clustering for the CBM

constructed from more finely grained real world objects (large scale). Such objects may be obtained by semantic generalization, aggregating other objects of a different nature that exist at a larger scale. This is the case of urban settlements that are seen as an aggregation of other objects (such as houses, streets, subways, and bridges), or woods that are made up of trees. The broad boundary of such objects represents a peripheral zone where the density of the elements composing the aggregate has an intermediate value.

Resulting from Variation of Context These correspond to linguistic propositions made up of a geographic location plus a qualitative modifier. The boundary of a region "the south of England" is dependent on the context in which the linguistic proposition is placed and its intended use. It might be a different region if the proposition is placed in a historic, architectural, or biological context; finally, it might even depend on different opinions of individuals. Also, the qualitative modifier, like all qualitative spatial terms, has a different meaning depending on the granularity with which it is defined. For example, in the two domains for qualitative modifiers (south, center, north) and (south, north), the proposition "the south of England" would subsume a rather different underlying region.

Future Directions

Current spatial database systems do not contemplate any explicit mechanism to handle uncertainty, with the exception of some metadata information, such as the resolution used to capture the data. The main problem about spa-

1	010100001	31	111111001	61	011011101	91	110011011	121	111110111
2	110100001	32	010100101	62	111011101	92	101011011	122	001001111
3	011100001	33	110100101	63	100111101	93	111011011	123	101001111
4	111100001	34	011100101	64	010111101	94	100111011	124	011001111
5	100010001	35	111100101	65	110111101	95	010111011	125	111001111
6	110010001	36	100010101	66	001111101	96	110111011	126	100101111
7	101010001	37	010010101	67	101111101	97	001111011	127	010101111
8	111010001	38	110010101	68	011111101	98	101111011	128	110101111
9	100110001	39	001010101	69	111111101	99	011111011	129	001101111
10	010110001	40	101010101	70	010100011	100	111111011	130	101101111
11	110110001	41	011010101	71	110100011	101	100100111	131	011101111
12	001110001	42	111010101	72	011100011	102	010100111	132	111101111
13	101110001	43	100110101	73	111100011	103	110100111	133	100011111
14	011110001	44	010110101	74	100010011	104	001100111	134	010011111
15	111110001	45	110110101	75	110010011	105	101100111	135	110011111
16	111001001	46	001110101	76	101010011	106	011100111	136	001011111
17	010101001	47	101110101	77	111010011	107	111100111	137	101011111
18	110101001	48	011110101	78	100110011	108	100010111	138	011011111
19	011101001	49	111110101	79	010110011	109	010010111	139	111011111
20	111101001	50	011001101	80	110110011	110	110010111	140	100111111
21	100011001	51	111001101	81	001110011	111	001010111	141	010111111
22	110011001	52	010101101	82	101110011	112	101010111	142	110111111
23	101011001	53	110101101	83	011110011	113	011010111	143	001111111
24	111011001	54	011101101	84	111110011	114	111010111	144	101111111
25	100111001	55	111101101	85	111001011	115	100110111	145	011111111
26	010111001	56	100011101	86	010101011	116	010110111	146	111111111
27	110111001	57	010011101	87	110101011	117	110110111		
28	001111001	58	110011101	88	011101011	118	001110111		
29	101111001	59	001011101	89	111101011	119	101110111		
30	011111001	60	101011101	90	100011011	120	011110111		

Objects with Broad Boundaries, Figure 7 The list of possible topological relations between uncertain lines

tial objects that are stored in spatial databases is that it is impossible to reason about the degree of indetermination of a result, since data have lost the information about all the sources of uncertainty resulting from various transformations that data have undertaken. An adequate modeling of geometric uncertainty in spatial data is essential for the assessment of data quality, and, consequently, for helping any process related to dataset acquisition and integration [11,13].

Broad boundaries record the information about uncertainty together with the data in order to be able to deal with it during any kind of spatial analysis. Future research directions are the definitions of extended spatial operators for objects with broad boundaries to support a spatial database implementation.

Cross References

- ► Approximation
- ► Dimensionally Extended Nine-Intersection Model (DE-9IM)
- ► Imprecision and Spatial Uncertainty
- ► Moving Object Uncertainty
- ► Representing Regions with Indeterminate Boundaries
- ► Spatial Data, Indexing Techniques
- ► Uncertainty, Modeling with Spatial and Temporal
- ► Vague Spatial Data Types

Recommended Reading

1. Clementini, E.: A model for uncertain lines. J Vis Language Comput **16**, 271–288 (2005)
2. Clementini, E., Di Felice, P.: An algebraic model for spatial objects with indeterminate boundaries. In: Burrough, P.A., Frank, A.U. (eds.) Geographic Objects with Indeterminate Boundaries, GISDATA Series, pp. 155–169. Taylor & Francis, London (1996)
3. Clementini, E., Di Felice, P.: A model for representing topological relationships between complex geometric features in spatial databases. Inf Sci **90**, 121–136 (1996)
4. Clementini, E., Di Felice, P.: Approximate topological relations. Int J Approx Reason **16**, 173–204 (1997)
5. Clementini, E., Di Felice, P.: A spatial model for complex objects with a broad boundary supporting queries on uncertain data. Data Knowl Eng **37**, 285–305 (2001)
6. Clementini, E., Di Felice, P., van Oosterom, P.: A small set of formal topological relationships suitable for end-user interaction. In: Abel, D., Ooi, B.C. (eds.) Advances in Spatial Databases – Third International Symposium, SSD '93. Lecture Notes in Computer Science, vol. 692, pp. 277–295. Springer, Berlin (1993)
7. Cohn, A.G., Gotts, N.M.: The 'egg-yolk' representation of regions with indeterminate boundaries. In: Burrough, P.A., Frank, A.U. (eds.) Geographic Objects with Indeterminate Boundaries, GISDATA Series, pp. 171–187. Taylor & Francis, London (1996)
8. Egenhofer, M.J., Al-Taha, K.: Reasoning about gradual changes of topological relationships. In: Frank, A.U., Campari, I., Formentini, U. (eds.) Theories and Models of Spatio-temporal Reasoning in Geographic Space. Lecture Notes in Computer Science, vol. 639, pp. 196–219. Springer, Berlin (1992)
9. Egenhofer, M.J., Herring, J.R.: Categorizing binary topological relationships between regions, lines, and points in geographic databases. Department of Surveying Engineering, University of Maine, Orono, ME (1991)
10. Erwig, M., Schneider, M.: Vague regions. In: Scholl, M., Voisard, A. (eds.) Advances in Spatial Databases – Fifth International Symposium, SSD '97. Lecture Notes in Computer Science, vol. 1262, pp. 298–320. Springer, Berlin (1997)

11. Frank, A.U.: Metamodels for data quality description. In: Jeansoulin, R., Goodchild, M. (eds.) Data Quality in Geographic Information – From Error to Uncertainty, pp. 15–29. Editions Hermès, Paris (1998)
12. Freksa, C.: Temporal reasoning based on semi-intervals. Artif Intell **54**, 199–227 (1992)
13. Guptill, S., Morrison, J.: Elements of spatial data quality. Elsevier, Oxford (1995)
14. Schneider, M.: Modelling spatial objects with undetermined boundaries using the Realm/ROSE approach. In: Burrough, P.A., Frank, A.U. (eds.) Geographic Objects with Indeterminate Boundaries, GISDATA Series, pp. 141–152. Taylor & Francis, London (1996)
15. Schneider, M.: Uncertainty management for spatial data in databases: fuzzy spatial data types. In: Guting, R.H., Papadias, D., Lochovsky, F. (eds.) Advances in Spatial Databases – Sixth International Symposium, SSD '99. Lecture Notes in Computer Science, vol. 1651, pp. 330–351. Springer, Berlin (1999)
16. Tøssebro, E., Nygård, M.: An advanced discrete model for uncertain spatial data. In: Meng, X., Su, J., Wang, Y. (eds.) Advances in Web-Age Information Management, Third International Conference (WAIM'02). Lecture Notes in Computer Science, vol. 2419, pp. 37–51. Springer, Berlin (2002)
17. Worboys, M., Clementini, E.: Integration of imperfect spatial information. J Vis Language Comput **12**, 61–80 (2001)

Oblique Images

► Photogrammetric Products

Octree

► Quadtree and Octree

OGC

► deegree Free Software
► Geography Markup Language (GML)
► Open Geospatial Consortium

OGC, OGC Simple Features

► Oracle Spatial, Geometries

OGC Standards, ODBC

► Smallworld Software Suite

OGC Web Service

► deegree Free Software

OGC Web Services

MARKUS LUPP
lat/lon GmbH, Bonn, Germany

Synonyms

OWS

Definition

Open Geospatial Consortium (OGC) Web Services (OWS) are services defined by the OGC, allowing all kinds of geospatial functionality. They include services for data access, data display and data processing. OWS requests are defined using the Hyper Text Transfer Protocol (HTTP) protocol and are encoded using key-value-pairs (KVP) structures or Extensible Markup Language (XML). The most widely known OWS is the Web Map Service (WMS).

Main Text

The currently most important distributed computing platform supported by the OGC is the web, or to be more precise the HTTP. HTTP defines two ways for passing information between clients and services, one of them being HTTP GET, most commonly associated with the KVP encoding and HTTP POST, usually used in conjunction with XML-encoded requests.

They can be classified into application services, portrayal services, data services, registry services and processing services [1].

Application services define interfaces for human interaction. Humans use application services to access portrayal, data, registry and processing services.

Portrayal (or visualization) services allow the display of geospatial data. Examples for such services are the Web Map Service (WMS) for display of maps and the Web Terrain Services (Web Perspective View Service) for display of three-dimensional data such as digital building or terrain models.

Data (access) services can be used for accessing the original geospatial data. Examples are Web Feature Service (WFS), for access of geospatial features encoded in GML and Web Coverage Service (WCS) for access to geospatial data describing space-varying phenomena such as satellite imagery, digital elevation models or triangulated irregular networks (TIN). There are also more specialized data access services, for example the Sensor Observation Service (SOS) for managing deployed sensors and retrieving sensor and specifically "observation" data.

Registry (or catalog) services define mechanisms to classify, register, describe, search, maintain and access information about geospatial resources available in a network. The

Catalog Service-Web (CS-W) specification and its profiles is the currently most important implementation specification of this category.

Processing services are used for processing geospatial data, for example to execute geometric calculations or transformation of coordinate reference systems. The Web Processing Service (WPS) is currently the most prominent example of a processing service.

Cross References

► deegree Free Software

References

1. Percivall G (2003) OpenGIS Reference Model. OGC project document 03-040. http://portal.opengeospatial.org/files/?artifact_id=3836

OGC's Open Standards for Geospatial Interoperability

GEORGE PERCIVALL
Open Geospatial Consortium, Herndon, VA, USA

Synonyms

Open standards; Web map service; WMS; Web coverage service; WCS; Standards; Interoperability; Exchange, data

Definition

This chapter summarizes the most significant aspects of the OGC (Open Geospatial Consortium) web services (OWS) architecture. This architecture is a service-oriented architecture, with all components providing one or more services to other services or to clients.

The definition of services includes a variety of applications with different levels of functionality to access and use geographic information. While specialized services will appropriately remain an area for proprietary products, standardization of the interfaces to those services allows interoperability between proprietary products. Geographic information system and software developers will use these standards to provide general and specialized services that can be used for all geographic information. The approach of this Chapter and the referenced standards is integrated with the approaches being developed within the more general world of information technology.

Historical Background

The OGC is an international industry consortium of companies, government agencies, research organizations, and universities participating in a consensus process to develop publicly available interface specifications. OpenGIS Standards support interoperable solutions that "geo-enable" the Web, wireless and location-based services, and mainstream IT. In the early 1990's OGC defined a vision for network based geospatial computing. Most recently this vision has come to fruition using web services. This section provides the vision from the 1990's followed by later sections that define the OGC Web Services architecture.

The widespread application of computers and use of geographic information systems (GIS) have led to the increased analysis of geographic data within multiple disciplines. Based on advances in information technology, society's reliance on such data is growing. Geographic datasets are increasingly being shared, exchanged, and used for purposes other than their producers' intended ones. GIS, remote sensing, automated mapping and facilities management (AM/FM), traffic analysis, geopositioning systems, and other technologies for Geographic Information (GI) are entering a period of radical integration.

Standards for geospatial interoperability provide a framework for developers to create software that enables users to access and process geographic data from a variety of sources across a generic computing interface within an open information technology environment.

- "a framework for developers" means that the International Standards are based on a comprehensive, common (i. e., formed by consensus for general use) plan for interoperable geoprocessing.
- "access and process" means that geodata users can query remote databases and control remote processing resources, and also take advantage of other distributed computing technologies such as software delivered to the user's local environment from a remote environment for temporary use.
- "from a variety of sources" means that users will have access to data acquired in a variety of ways and stored in a wide variety of relational and non-relational databases.
- "across a generic computing interface" means that standard interfaces provide reliable communication between otherwise disparate software resources that are equipped to use these interfaces.
- "within an open information technology environment" means that the standards enable geoprocessing to take place outside of the closed environment of monolithic GIS, remote sensing, and AM/FM systems that control and restrict database, user interface, network, and data manipulation functions.

Scientific Fundamentals

The fundamentals principles of the OGC Web Services (OWS) architecture, include:

a) Service components are organized into multiple tiers.
   1) All components provide services, to clients and/or other components, and each component is usually called a service (with multiple implementations) or a server (each implementation).
   2) Services (or components) are loosely arranged in four tiers, from Clients to Application Services to Processing Services to Information Management Services, but un-needed tiers can be bypassed.
   3) Services can use other services within the same tier, and this is common in the Processing Services tier.
   4) Servers can operate on (tightly bound) data stored in that server and/or on (loosely bound) data retrieved from another server.
b) Collaboration of services produce user-specific results.
   1) All services are self-describing, supporting dynamic (just-in-time) connection binding of services supporting publish-find-bind.
   2) Services can be chained with other services and often are chained, either transparently (defined and controlled by the client), translucently (predefined but visible to the client), and opaquely (predefined and not visible to client), see Subclause 7.3.5 of [ISO 19119]
   3) Services are provided to facilitate defining and executing chains of services.
c) Services communication uses open Internet standards.
   1) Communication between components uses standard World Wide Web (WWW) protocols, namely HTTP GET, HTTP POST, and SOAP.
   2) Specific server operations are addressed using Uniform Resource Locators (URLs).
   3) Multipurpose Internet Mail Extensions (MIME) types are used to identify data transfer formats.
   4) Data transferred is often encoded using the Extensible Markup Language (XML), with the contents and format specified using XML Schemas.
d) Service interfaces use open standards and are relatively simple.
   1) OGC web service interfaces are coarse-grained, providing only a few static operations per service.
   2) Service operations are normally stateless, not requiring servers to retain interface state between operations.
   3) One server can implement multiple service interfaces whenever useful.
   4) Standard XML-based data encoding languages are specified for use in data transfers.
e) Server and client implementations are not constrained.
   1) Services are implemented by software executing on general purpose computers connected to the Internet. The architecture is hardware and software vendor neutral.
   2) The same and cooperating services can be implemented by servers that are owned and operated by independent organizations.
   3) Many services are implemented by standards-based Commercial Off The Shelf (COTS) software.

Key Applications

Services Tiers

Except for clients, all OWS architecture components provide services, to clients and/or to other components. Each such component is usually called a service when multiple implementations are expected, and each implementation is called a server (or service instance). These components are thus usually called services or servers in this chapter.

Clients are software packages that provide access to a human user, or operate as agents on behalf of other software. Software that provides access to a human user can be thin (e. g., a web browser), thick (a large application), or "chubby" (in between).

All services (or components) are loosely organized in four tiers.

- Client tier
- Application Services tier
- Processing Services tier
- Information Management Services tier

This organization is loose in that clients and services can bypass un-needed tiers, as indicated by some arrows. Services can use other services within the same tier, and this is common especially in the Processing Services tier. Also, some services perform functions of more than one tier, when those functions are often used together and combined implementation is more efficient. Assignment of such combined services to tiers is somewhat arbitrary.

This OWS architecture is designed for use where data is important and often voluminous. Servers can operate on (tightly bound) data stored in that server and/or on (loosely bound) data retrieved from another server. Most data is stored by the servers in the Information Management Services tier, but some data (can be and often) is stored in other services and servers.

Application Services Tier The Application Services tier contains services designed to support Clients, especially thin client software such as web browsers. That is, these Application Services are designed for use by clients instead of each client directly performing these often-needed support functions. The services in the Application Services tier are used by Clients, and can use other services in the Application Services, Processing Services, and

OGC's Open Standards for Geospatial Interoperability, Table 1 Some specific Application Services

Service name	Service description
Web portal services	Services that allow a user to interact with multiple application services for different data types and purposes
WMS application services	Services that allow a user to interact with a Web Map Service (WMS) to find, style, and get data of interest
Geographic data extraction services	Services that allow a user to extract and edit feature data, interacting with images and feature data
Geographic data management services	Services that allow a user to manage geospatial data input and retirement, interacting with Information Management Services
Chain definition services	Services to define a service chain and enable it to be executed by the workflow enactment service; may also provide a chain validation service
Workflow enactment services	Services to interpret chain definitions and control instantiation of servers and sequencing of activities, maintaining internal state information associated with various services being executed

OGC's Open Standards for Geospatial Interoperability, Table 2 Some specific Processing Services

Service name	Service description
Web Coordinate Transformation Service (WCTS)[b]	Transforms the coordinates of feature or coverage data from one coordinate reference system (CRS) to another, including "transformations", "conversions", rectification, and orthorectification
Web Image Classification Service (WICS)	Performs classification of digital images, using client-selected supervised or unsupervised image classification method
Feature Portrayal Service (FPS)	Dynamically produces client-specified pictorial renderings in an image or graphics format of features and feature collections usually dynamically retrieved from a Web Feature Server (WFS)
Coverage Portrayal Service (CPS)	Dynamically produces client-specified pictorial renderings in an image or graphics format of a coverage subset dynamically retrieved from a Web Coverage Service (WCS)
Geoparser Service	Service to scan text documents for location-based references, such as a place names, addresses, postal codes, etc., for passage to a geocoding service
Geocoder Service	Service to augment location-based text references with position coordinates
Dimension measurement services	Services that compute dimensions of objects visible in an image or other geospatial data
Route determination services	Determine optimal path between two specified points based on input parameters and properties contained in a Feature Collection; may also determine distance between points and/or time to follow path
Change detection services	Services to find differences between two data sets that represent the same geographical area at different times
Feature generalization services	Service that reduces spatial variation in a feature collection to counteract the undesirable effects of scale reduction
Format conversion services	Service that converts data from one format to another, including data compression and decompression

Information Management Services tiers. The specific services included in this tier include (but are not limited to) the services listed in Table 1.

Processing Services Tier The Processing Services tier contains services designed to process data, sometimes both feature and image (coverage) data. The services in the Processing Services tier are used by clients and by services in the Application Services tier. These services can use other services in the Processing Services and Information Management Services tiers. The specific services included in this tier include (but are not limited to) the services listed in Table 2.

Information Management Services Tier The Information Management Services tier contains services designed to store and provide access to data, normally handling multiple separate datasets. In addition, metadata describing multiple datasets can be stored and searched. Access is usually to retrieve a client-specified subset of a stored dataset, or to retrieve selected metadata for all datasets whose metadata meets client-specified query constraints.

The services in the Processing Services tier are used by clients and by services in the Application Services and Processing Services tiers. These services can use other services in the Information Management Services tier. The

OGC's Open Standards for Geospatial Interoperability, Table 3 Some specific Information Management Services

Service name	Service description
Web Map Service (WMS)	Dynamically produces spatially referenced maps of client-specified ground rectangle from one or more client-selected geographic datasets, returning pre-defined pictorial renderings of maps in an image or graphics format
Web Feature Service (WFS)	Retrieves features and feature collections stored that meet client-specified selection criteria
Web Coverage Service (WCS)	Retrieves client-specified subset of client-specified coverage (or image) dataset
Catalog Service for the Web (CSW)	Retrieves object metadata stored that meets client-specified query criteria
Order handling services	Allows clients to order products from a provider, including: selection of geographic processing options, obtaining quotes on orders, submission of order, statusing of orders, billing, and accounting

specific services included in this tier include (but are not limited to) the services listed in Table 3.

Service Trading (Publish – Find – Bind)

All OGC architecture services are self-describing, supporting dynamic (just-in-time) connection binding of servers using service trading. Service trading addresses discovery of available service instances. Trading facilitates the offering and the discovery of interfaces which provide services of particular types. A trader implementation records service offers and matches requests for advertised services. Publishing a capability or offering a service is called "export". Matching a service request against published offers or discovering services is called "import". This can also be depicted in an equivalent manner as the "Publish – Find – Bind" (PFB) pattern of service interaction. The fundamental roles are:

a) Trader (Registry) – registers service offers from exporter objects and returns service offers to importer objects upon request according to some criteria.
b) Exporter (Service) – registers service offers with the trader object
c) Importer (Client) – obtains service offers, satisfying some criteria, from the trader object.

NOTE In the OWS architecture, a Registry is implemented using the Catalog Service for the WEB (CSW) service interface.

A trader plays the role of "matchmaker" in a service-oriented architecture. The interaction pattern is:

a) To publish a service offer, an Exporter gives a Trader a description of a server, including a description of the interface at which that service instance is available.
b) To find suitable server offers, an Importer asks a Trader for a server having certain characteristics. The trader checks the previously registered descriptions of servers, and responds to the importer with the information required to bind with a server. Preferences may be applied to the set of offers matched according to service type, constraint expressions, and various policies. Use of preferences can determine the order used to return matched offers to the importer.
c) To bind a service, an Importer applies information received from the Trader to bind to a server. The Client then proceeds to use that server.

Service Chaining

In many cases, multiple services must be used together to perform a useful function. The OWS architecture thus supports "chaining" together of multiple servers, and such chaining is frequently used. This chaining is not limited to a linear chain; a network of services can also be "chained". Within such a chain, most servers input the data that is output from the previous server in the chain. Services can be chained transparently (defined and controlled by the client), translucently (predefined but visible to the client), and opaquely (predefined and not visible to client), see Subclause 7.3.5 of [OGC 02-006, ISO 19119].

To facilitate service chaining, some services are defined to support defining and executing chains of services. Also, some Processing Service interfaces are designed to support retrieving the data to be processed from another service, which can be an Information Management Service or another Processing Service.

To allow more efficient execution of server chains, some service interfaces support server storage of operation results until requested by next service in a chain. This approach separates the flow of control from the flow of data.

Service Communication

Communication between clients and services, and between services, uses only open non-proprietary Internet standards. That is, the OWS architecture uses the Internet or equivalent as its distributed computing platform (DCP). More specifically, communication between components uses standard World Wide Web (WWW) protocols, name-

OGC's Open Standards for Geospatial Interoperability, Table 4 Some standardized encoding formats and languages

Specification name	Description
Styled Layer Descriptor (SLD)	Encodes client-controlled styling for map portrayal of features and coverages (images)
Geography Markup Language (GML)	Language defined using XML Schemas based on the ISO 191XX series of standards, to be used to specify application-specific XML Schemas
Coordinate Reference Systems (part of GML)	Encodes definitions of coordinate reference systems, coordinate systems, datums, and coordinate transformations (and conversions)
OWS Context	Encodes multiple OWS application display context
URNs using ogc URN namespace	Standardized Universal Resource Identifiers (URNs) referencing most well-known coordinate reference systems (CRSs) and grid CRSs
Web Service Description Language (WSDL)	Encodes web service interfaces
Business Process Execution Language (BPEL)	Encodes process sequences for specific purposes

ly HTTP GET, HTTP POST, and Simple Object Adaptor Protocol (SOAP). Specific operations of specific servers are addressed using Uniform Resource Locators (URLs). Multipurpose Internet Mail Extensions (MIME) types are used to identify data transfer formats. The data transferred is often encoded using the Extensible Markup Language (XML), with the contents and format carefully specified using XML Schemas.

Service Interfaces

OGC web service interfaces use open standards and are relatively simple. All services support open standard interfaces from their clients, often OGC-specified service interfaces. In addition to being well-specified and interoperable tested, the OGC-specified service interfaces are coarse-grained, providing only a few static operations per service. For many services, only three service operations are specified. One server can implement multiple service interfaces whenever useful.

The OGC web service interfaces are usually stateless, so session information is not passed between a client and server. Clients retain any needed interface state between operations.

The OGC web service interfaces share common parts whenever practical, allowing those parts to be specified and implemented only once. For example, all OWSs have a mandatory GetCapabilities operation to retrieve server metadata. That server metadata includes four required sections, with the contents and format of three sections common to all services, and part of the fourth section common to most services. In addition, many service interfaces have multiple specified levels of functional compliance, or multiple specialized subset and/or superset profiles.

Standard XML-based data encoding formats and languages are used in many server-to-client and client-to-server data transfers. The formats and languages specified include (but are not limited to) those listed in Table 4. In these formats and languages and elsewhere, the geographic data and service concepts are closely based on the ISO 191XX series of standards.

Server Implementation

Servers and client implementations are not constrained except for supporting the specified service interfaces. Each can be implemented by software executing on any general purpose computer connected to the Internet or equivalent. The architecture is hardware and software vendor neutral. The same and cooperating services can be implemented by servers that are owned and operated by independent organizations.

All OWS services and clients are implemented by available standards-based Commercial Off The Shelf (COTS) software. This commercial software can sometimes be used without requiring major software development, or can be adapted to specific needs with limited software development. Software may be developed as proprietary or open source code.

Future Directions

Existing OGC specifications are widely implemented in the marketplace. The specifications for the Information Management Tier (WMS, WFS, WCS, CSW) are mature with multiple implementations. Emphasis in the OGC development activities is focused on developing Processing Services and best practices for service chaining, e. g., workflow. Recently OGC adopted the Web Processing Service (WPS) specification as an approach to develop standard service interfaces for the numerous types of processing algorithms. The OWS architecture is also the basis for the recently developed OGC Sensor Web Enablement (SWE) set of standards for accessing any type of sensor as a web service. Approaches for managing digital rights in the OWS environment are being developed based on an approach that will not require modification to existing OWS services. Concepts are underdevelopment for applying the OWS architecture and services in a mass market environment.

Cross References

- ▶ Catalogue Information Model
- ▶ Computer Environments for GIS and CAD
- ▶ Cyberinfrastructure for Spatial Data Integration
- ▶ Geographic Coverage Standards and Services
- ▶ Geography Markup Language (GML)
- ▶ Information Services, Geography
- ▶ Internet GIS
- ▶ Internet-Based Spatial Information Retrieval
- ▶ Metadata and Interoperability, Geospatial
- ▶ Modeling with ISO 191xx Standards
- ▶ Open-Source GIS Libraries
- ▶ University of Minnesota (UMN) Map Server
- ▶ Web Feature Service (WFS)
- ▶ Web Feature Service (WFS) and Web Map Service (WMS)
- ▶ Web Mapping and Web Cartography
- ▶ Web Services, Geospatial

Recommended Reading

1. OGC 00-014r1, Guidelines for Successful OGC Interface Specifications
2. OGC 2001, Introduction to OGC Web Services
3. OGC 02-006, OpenGIS Abstract Specification Topic 12: OpenGIS Service Architecture
4. OGC 03-025, OpenGIS Web Services Architecture
5. OGC 05-008, OGC Web Services Common Specification, version 1.0.0
6. OGC 05-010, URNs of definitions in ogc namespace
7. OpenGIS® Standards: http://www.opengeospatial.org/standards

OGIS

- ▶ Open-Source GIS Libraries
- ▶ PostGIS

OGR

- ▶ Open-Source GIS Libraries

OLAP Operations

- ▶ OLAP, Spatial

OLAP Query

- ▶ Top-k OLAP Queries Using Augmented Spatial Access Methods

Olap Results, Distributed Caching

PANOS KALNIS, GABRIEL GHINITA
Department of Computer Science,
National University of Singapore, Singapore, Singapore

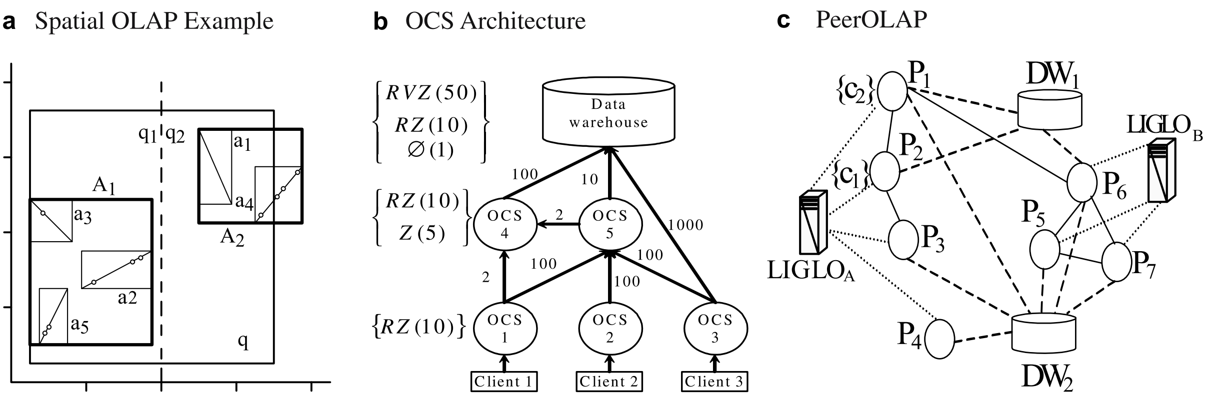
Synonyms

Peer-to-peer caching for spatial on-line analytical processing; Caching; Distributed caching

Definition

Spatial databases store information about the position of individual objects in space. In many applications, such as traffic supervision or mobile communications, only summarized data, like the number of cars in an area, or number of phones serviced by a cell, is required. Although this information can be obtained from transactional spatial databases, its computation is expensive, rendering online processing inapplicable. Driven by the nonspatial paradigm, spatial data warehouses (DW) can be constructed to accelerate spatial on-line analytical processing (OLAP) operations. Typically, DWs employ precomputation and caching of previous results [1,2]; these methods are centralized.

Here, the focus is on *distributed* techniques for caching the results of spatial OLAP queries. Users are assumed to be far away (in terms of network cost) from the DW; it is therefore beneficial to access cached data that are located physically closer to them. The intuition is similar to using proxy servers in order to accelerate the downloading of web pages.An OLAP caching technique that works in conjunction with existing proxy servers for web pages is introduced in [3]. The DW sends an applet together with the OLAP results, which is executed on the proxy server and coordinates the caching of OLAP data. A different approach, which assumes a dedicated infrastructure of OLAP Cache servers (OCSs) is presented in [4]. The infrastructure consists of a set of geographically spanned OCSs, which dynamically cache results from OLAP queries with the objective of minimizing query response time. The structure of the OCS network is determined by the administrators and is fixed during the lifetime of the system. On the other hand, [5] proposes PeerOLAP, a protocol that integrates distributed OLAP caching



Olap Results, Distributed Caching, Figure 1 Spatial on-line analytical processing (*OLAP*) caching architectures. *OCS* OLAP cache servers, *LIGLO* location independent global name lookup, *DW* data warehouses

with peer-to-peer (P2P) technology. In PeerOLAP the system resources are not known in advance; instead, caching peers can enter and leave the network at any time. Therefore, PeerOLAP implements methods to locate resources and dynamically restructure the network. Also, compared to previous systems, PeerOLAP allows caching at a finer granularity.

Historical Background

The need for decision support systems is of paramount importance in today's business, leading many enterprises to build specialized Data Warehouses (DWs). Decision makers issue OLAP (as opposed to transactional) queries that typically involve aggregations of millions of rows, in order to identify interesting trends. Users perceive the data of the DW as cells in a multidimensional data cube [6]. Fetching from the DW the necessary parts of the cube and performing aggregation, is an extremely time-consuming task. A common technique to accelerate such queries is to precalculate and store some results. Such stored fragments are essentially parts of views in relational database terms; their storage is referred to here as *materialization* or *caching* of OLAP views. Most of the existing work on static and dynamic view materialization [1,2,6] is limited to a central server that stores views and employs them to answer queries with minimal cost.

Centralized caching is not adequate for large-scale, distributed environments (e. g., the internet). Distributed caching is used as a primary technique for coping with high latency experienced by end-users in the web. There are four major locations where caching is performed [7]: proxy at the front-end of a server farm, network cache at the end-points of the backbone network, LAN proxy and web browser. Although caching at these locations has been shown to significantly reduce web traffic, dynamically generated pages, consisting of a static part and a dynamic part (for example, query results from a database with a web server linked to it), are not suitable for page-level caching. To facilitate distributed caching of OLAP results, recent research [3,4,5] has focused on extending the caching algorithms of proxy servers or implementing specialized infrastructures, alternatives looked at in detail below.

Scientific Fundamentals

Consider a spatial DW consisting of a fact table that stores the coordinates of vehicles. The DW may have several "traditional" dimensions, such as the type of the vehicle, the time dimension, etc. Additionally, there can be spatial dimensions. For example, the Roads dimension can store the spatial representation of each road (i. e., polyline) and impose various levels of hierarchy (e. g., roads are grouped in zones around police stations); Fig. 1a presents an example. A typical spatial OLAP query would ask for the number of vehicles grouped by zone, time and type of vehicle; the user could then drill-down to see the number of vehicles on each road of a specific zone.

In terms of caching, we do not distinguish between spatial and nonspatial dimensions. Therefore, the various combinations of grouping attributes form a partial ordering (i. e., a lattice, following the formulation from [6]). The examples in the rest of this article assume three abstract dimensions R, V, Z which may be either spatial or nonspatial.

OLAP Caching in Proxy Servers

A method of caching both web pages and OLAP query results in common proxy servers, using the existing web proxy infrastructure is proposed in [3]. Practical applications are expected to embed OLAP data inside web pages.

For example, a web page may include a map to show the real-time traffic conditions at major roads of a city. Proxy servers are enhanced with an active caching mechanism that allows them to dynamically generate requested pages based on views cached from previous queries. A control applet is kept together with the static part of the page and in the presence of a request the applet fetches the dynamic data from the original site and combines them with the cached static part to create a hypertext markup language (HTML) document. An analytical cost model of query processing and a cache replacement strategy are introduced, both based on the *popularity-aware greedy-dual-size* algorithm [8]. The experimental evaluation of the hybrid WWW-OLAP caching infrastructure shows that it is considerably superior to the traditional web proxy infrastructure, even if the proportion of OLAP queries is as low as 30% of the combined web and OLAP workload.

OLAP Cache Servers

The *OLAP Cache Servers* (OCS) [4] architecture is similar to the proxy-server architecture, which is widely used to cache web pages. However, it forms a specialized infrastructure for caching results of OLAP queries. The architecture consists of three layers (see Fig. 1b): the DW layer, the OCS layer and the client layer. In the general case, there are many unrelated warehouses, corresponding to different enterprises. The DWs are connected to a number of OCSs, which are interconnected through an arbitrary network. The clients do not access the warehouses directly; instead, they send their queries to an OCS. Each OCS contains a cache where previous results are stored and has computational capabilities, so it can use the cached results for further aggregations. The decision on which results to cache is taken autonomously by the OCSs. The intuition behind this architecture is that most of the time, some OCS in the network will be able to satisfy the requests from the clients, saving network and computation cost incurred by directing the request to the data warehouse.

As an example, consider the configuration of Fig. 1b. There are three clients, a network of five OCSs and a DW. The cache contents are presented next to the OCSs, and the numbers in parenthesis represent the size of the views. The edges correspond to network connections; there is a directed edge from a site i to a site j, if i can direct a query to j. The numbers next to the edges denote the network cost. For simplicity, assume that each client is connected to only one OCS and the network cost between a client and its corresponding OCS is zero.

Assume that $Client_1$ asks a query $q = \varnothing$ (i. e., an aggregation query, without a GROUP BY clause). OCS_1 checks the query and estimates that it can calculate $\varnothing$ from RZ with cost 10, but it also contacts its first-degree neighbors. OCS_5 cannot calculate the answer. On the other hand, OCS_4 can calculate q from Z with cost 5 and transfer the result back with cost 2 (i. e., the total cost is 7); therefore, the query is redirected to OCS_4. However, since OCS_4 does not have the exact answer, it asks the DW, which has already materialized $\varnothing$, but due to a slower network, the total cost is $1 + 100 = 101$. So OCS_4 calculates locally the result and sends it to OCS_1. At the same time OCS_4 decides whether it should cache the new result at its local cache, based on a *goodness* metric. This metric takes into consideration the requests at various levels of the lattice, the cost of recomputing the result and the corresponding network cost. Similarly, OCS_1 returns the result to the client and decides whether to store the result in its own cache.

When the data in the warehouse are updated, the cached results in the OCSs are not invalidated. Instead, the warehouse calculates the necessary deltas and when a result is required, the corresponding OCS decides whether to ask for the updated view, or to request the corresponding delta and incrementally update the cached data. Actually, the deltas themselves are objects just like the query results, and are cached in the OCSs using the same mechanism.

PeerOLAP

P2P technology has recently attracted a lot of attention, since it allows the implementation of large distributed repositories of digital information. In a P2P system, numerous nodes of equal roles are connected through an arbitrary network and exchange data or services directly with each other.

The PeerOLAP [4] architecture implements a P2P system where each user corresponds to a peer. Users pose OLAP queries and at the same time they share their cache space with all other peers. To provide an entry point to the network, *location independent global name lookup* (LIGLO) servers are employed to maintain a list of online peers, together with details about the warehouses that they access. Each peer P_i has a local cache and implements a mechanism for publishing its cache contents and its computational capabilities. The local cache contains *fragments of views* (i. e., *chunks*); chunks from different peers can be combined to create an OLAP query result. Other peers can connect to P_i and request a result. P_i may either answer the query (or part of it) locally, if it has the required data, or propagate the query to its neighbors. In either case, all results return directly to the peer that initiated the query. The goal of PeerOLAP is to act as a combined virtual cache, where all the components offer resources to minimize the query cost.

Figure 1c depicts a typical PeerOLAP network consisting of seven peers and two data warehouses. There is an arbitrary set of connections among peers denoted by solid lines, and each peer also connects directly to one or multiple warehouses simultaneously. Assume that P_2 issues a query q which requires chucks c_1, c_2 and c_3 (i. e., a GROUP BY query with a WHERE clause). If c_1 is already at the local cache, P_2 will send a request for c_2 and c_3 to its neighbors P_1 and P_3. P_1 contains c_2, therefore it estimates the cost of retrieving and transferring this result back to P_2, and at the same time it forwards the request to P_6. Note that both c_2 and c_3 are requested since P_6 may be able to provide c_2 with lower cost than P_1.

In order to avoid flooding the network with messages, a maximum number of hops is assigned to each message. Assuming that this number is 2, the query will not be propagated to the neighbors of P_6. On the other hand, P_3 will not forward the message although there is still one hop allowed, since a peer can direct to the DW only its local queries in order to avoid overloading the DW with the same message. There is also a mechanism for breaking message loops: each peer keeps a list of recent messages and rejects the ones that had been processed before.

P_2 does not know how many peers will respond. Therefore, it waits until all requested chunks are found or a timer expires. Missing chunks are requested from the DW. Note that although the warehouse can provide any chunk, it is the last option due to the high network cost.

After the query has terminated, P_2 decides which chunks to keep in the local cache. Each chunk is assigned a *benefit* value. If an incoming chunk c has a higher benefit than some cached results, these results are evicted and c is stored, else c is rejected. For chunks that were sent directly from the warehouse (meaning that they were not found in the neighborhood of P_2), PeerOLAP explores the option of caching them in some neighbor of P_2 (i. e., P_1 or P_3). This characteristic differentiates PeerOLAP from similar systems, since it treats the cache space of all peers as a large virtual cache, instead of small noncooperative individual caches.

PeerOLAP incorporates a network reorganization mechanism that aims at optimizing the network structure, by creating virtual neighborhoods of peers with similar query patterns. The goal is to assign a set of neighbors to each peer P, so that there is a high probability for P to obtain missing chunks directly from its neighbors, without having to search a large part of the network. The available network resources of each peer are modeled as a second level of caching, where the cached objects correspond to open TCP/IP connections to other peers. Each connection is assigned a benefit value and the most beneficial connections are selected to be the peer's neighbors. The experimental evaluation of PeerOLAP shows that good results can be obtained even with a limited set of beneficial neighbors.

Compared to the OCS architecture, there are two major differences:

- The set of OCSs is a dedicated infrastructure, where the network structure is determined when each OCS enters the network and remains fixed thereafter. On the other hand, PeerOLAP connects a large number of client computers (i. e., peers) into a virtual cache. Peers are allowed to enter and leave the network at any time. Therefore, PeerOLAP implements mechanisms to locate resources and dynamically optimize the network structure.
- OCSs cache only entire views, corresponding to nodes in the datacube lattice, whereas PeerOLAP caches *fragments* of views (i. e., chunks). Owing to the finer granularity, PeerOLAP has the flexibility to combine data from many peers simultaneously, in order to answer a query; therefore, the probability of answering a query from the cache is maximized.

Key Applications

Distributed caching is complementary to many spatial OLAP applications, which allow numerous users to access the DW through a wide-area network (e. g., the internet). Below, we present some examples:

Traffic Monitoring

A spatial OLAP traffic supervision system would monitor the positions of cars in a city and the road traffic. An OLAP query in the system would be: "find the road segments with the heaviest traffic near the center" or, given a medical emergency, "which is the hospital that can be reached faster given the current traffic conditions." In both cases, it is statistical information, i. e., the number of cars, rather than their identification that is important.

Urban Planning

Urban planning application would require a spatial OLAP system to answer queries such as "find the number of buildings less than four stories high in the city center area" or "determine the suburban regions with the most accentuated growth in private residences."

Decision Support in Cellular Networks

Different GSM network cells may exhibit a large variation in base station load. A spatial OLAP system would help identifying the most loaded cells and decide where new base stations must be deployed.

Environment Preserving

Environmental organizations make intensive efforts to keep track of endangered species, or to monitor pollution levels. A spatial OLAP system may help answering queries such as "find the number of trees from an endangered species that are within 10 miles of a mine exploitation facility."

Future Directions

In the near future we expect to see more distributed caching systems based on the P2P technology, since they are inexpensive and easy to deploy. Similarly to PeerOLAP, we expect these systems to have some mechanism to optimize the network structure. More sophisticated approaches (compared to PeerOLAP) are possible, which will consider factors such as the geographical location of peers and statistical information about each peer's interests. Furthermore, PeerOLAP is based on an *unstructured* P2P architecture. Future systems may be implemented on top of structured P2P systems, such as Chord or CAN.

Cross References

► Aggregation Query, Spatial
► Data Analysis, Spatial

Recommended Reading

1. Kotidis, Y., Roussopoulos, N.: DynaMat: A dynamic view management system for data warehouses. In: Proceedings of ACM-SIGMOD International Conference on Management of Data, Philadelphia, Pennsylvania, USA, 1–3 June 1999, pp. 371–382
2. Scheuermann, P., Shim, J., Vingralek, R.: WATCHMAN: A data warehouse intelligent cache manager. In: Proceedings of 22nd International Conference on Very Large Data Bases (VLDB), Mumbai, India, 3–6 Sept 1996, pp. 51–62
3. Loukopoulos, T., Kalnis, P., Ahmad, I., Papadias, D.: active caching of on-line-analytical-processing queries in WWW proxies. In: Proceedings of the 2001 International Conference on Parallel Processing (ICPP), Valencia, Spain, 3–7 Sept 2001, pp. 419–426
4. Kalnis, P., Papadias, D.: Proxy-Server Architectures for OLAP. In: Proceedings of ACM-SIGMOD International Conference on Management of Data, Santa Barbara, California, USA, 21–24 May 2001, pp. 367–378
5. Kalnis, P., Ng, W.S., Ooi, B.C., Papadias D., Tan K.L.: Proxy-server architectures for OLAP. In: Proceedings of ACM-SIGMOD International Conference on Management of Data, Madison, Wisconsin, 3–6 June 2002, pp. 25–36
6. Harinarayan, V., Rajaraman, A., Ullman, J.D.: Implementing data cubes efficiently. In: Proceedings of ACM-SIGMOD International Conference on Management of Data, Montreal, Quebec, Canada, 4–6 June 1996, pp. 205–216
7. Bestavros, A.: WWW traffic reduction and load balancing through server-based caching. IEEE Concurrency: Spec. Issue Parallel Distrib. Technol. **5**, pp. 56–67 (1997)
8. Jin, S., Bestavros, A.: Popularity-Aware Greedy Dual-Size Web Proxy Caching Algorithms. In: Proceedings of the International Conference on Distributed Computing Systems (ICDCS), Taipei, Taiwan, 10–13 Apr 2000, pp. 254–261

OLAP, Spatial

JIAWEI HAN
Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA

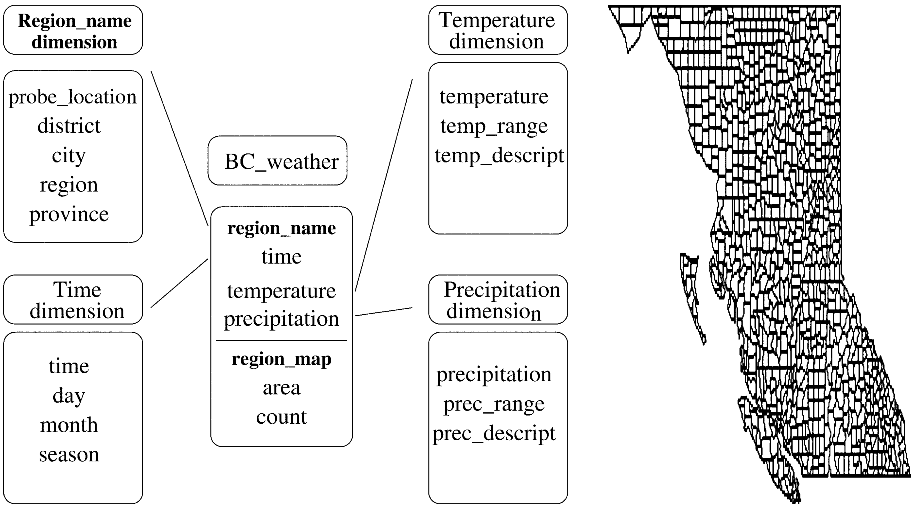
Synonyms

Spatial online analytical processing; Spatial multidimensional analysis; Spatial data warehouses; Data cube; OLAP operations

Definition

OLAP is an abbreviation of **O**n-**L**ine **A**nalytical **P**rocessing, that refers to a set of online (*i. e.*, fast and interactive) operations, such as *drilling, rolling, dicing, slicing*, and *pivoting*, exploring a set of predefined aggregates (e. g., *sum, average, count*, and *variance*) of some measures (e. g., temperature, area, and population) in a multidimensional data warehouse [2]. Spatial OLAP refers to such OLAP operations in a spatial data warehouse, such as drilling into the detailed spatial locations (e. g., finding detailed fact summaries in a city district), or slicing of a data cube with a set of constraints (e. g., finding average midnight temperature distribution in May in the city of Chicago). It is desirable to support spatial OLAP operations in spatial data warehouses since a user can interactively explore a multidimensional spatial data warehouse at his/her finger tips. A spatial data warehouse needs to first integrate spatial information from multiple spatial databases collected in different organizations, at different times, and of multiple scales. Dimensions defined for such spatial data warehouses should include not only traditional dimensions on time, location, subject, organization, etc. but also spatial dimensions, such as maps in multiple scales, from fine to rough. Summarized maps can also be taken as spatial measures. Spatial data cubes can be constructed on such multidimensional spatial warehouse. Spatial OLAP can then be performed on such spatial data cubes for efficient and flexible interactive multidimensional data exploration.

For example, spatial OLAP can be used for the analysis of regional weather patterns. Suppose there are about 3,000 weather probes scattered in British Columbia (BC), each recording daily temperature and precipitation for a designated small area and transmitting signals to a provincial



OLAP, Spatial, Figure 1 Star model of a spatial data warehouse: BC_weather and corresponding BC weather probes map

weather station. A user may like to view weather patterns on a map by month, by region, and by different combinations of temperature and precipitation, or may like to dynamically drill-down or roll-up along any dimension to explore desired patterns, such as wet and hot regions in Fraser Valley in July, 2006. A star model can be constructed, as shown in Fig. 1, for this *BC_weather* warehouse, with four dimensions: *temperature, precipitation, time*, and *region_name*, and three measures: *region_map, area*, and *count*. A concept hierarchy for each dimension can be created by users or experts or generated automatically by data clustering or data analysis. Table 1 presents the hierarchies for dimensions in *BC_weather* warehouse. Of the three measures, *region_map* is a *spatial* measure which represents a collection of spatial pointers to the corresponding regions, *area* and *count* are *numerical* measures, representing the sum of the total areas of the corresponding spatial objects and the total number of base regions (probes) accumulated in the corresponding cell, respectively. With these dimensions and measures, spatial OLAP operations can be performed by stepping up and down along any dimension presented in Fig. 1.

Historical Background

Spatial OLAP is developed on top of several fundamental concepts and technologies in database systems [4], spatial database systems [13], data warehouse [2], and OLAP technology [5,7]. The research and development of database systems have generated rich technologies, including the modeling of large data sets as a set of interrelated data relations using a relational or entity-relationship data model, and the methods for storing, updating, and accessing data, such as indexing methods, including single-dimensional B+-trees, and multi-dimensional R-trees or quad-trees, query processing and optimization methods, and many other advanced techniques. By extending database systems to incorporate spatial data model, spatial query languages, and spatial accessing methods, spatial database systems manage a large amount of spatial and nonspatial data and support various kinds of large-scale spatial data applications [11]. Data warehouse and OLAP are the technologies that provide tools for integration of multiple data sources by data cleaning, data transformation, and heterogeneous data integration, and tools for precomputation, storage, indexing, and accessing of multidimensional aggregates for efficient online analytical processing. Spatial data warehouse [1] and spatial OLAP [9] are the technologies developed by integration of spatial database systems and data warehouse/OLAP technologies. They support spatial data integration and online analytical processing of massive, multidimensional spatial data.

Scientific Fundamentals

The scientific fundamentals of spatial OLAP come from several related technologies: spatial database systems, data warehouse systems, and OLAP technology. These systems and technologies have been extensively researched in the related domains. Although built on such technologies, spatial data warehouse and spatial OLAP encounter new challenges in research and implementation as illustrated below. First, the construction of spatial data warehouses needs the integration of spatial data from heterogeneous sources and systems. Spatial data is usually stored in different industry firms and government agencies using different data formats. Data formats are not only structure-specific (e. g., raster- vs. vector-based spatial data, object-oriented vs.

Region_name:	Time:
probe location ⊂ district ⊂ city ⊂ region ⊂ province	hour ⊂ day ⊂ month ⊂ season
Temperature:	Precipitation:
any ⊃ (cold, mild, hot)	any ⊃ (dry, fair, wet)
cold ⊃ (below −20, −20 to −10, −10 to 0)	dry ⊃ (0 to 0.05, 0.05 to 0.2)
mild ⊃ (0 to 10, 10 to 15, 15 to 20)	fair ⊃ (0.2 to 0.5, 0.5 to 1.0, 1.0 to 1.5)
hot ⊃ (20 to 25, 25 to 30, 30 to 35, above 35)	wet ⊃ (1.5 to 2.0, 2.0 to 3.0, 3.0 to 5.0, above 5.0)

OLAP, Spatial, Table 1
Hierarchy for each dimension in *BC_weather*

relational models, different spatial storage and indexing structures, etc.), but also vendor-specific (e. g., ESRI, MapInfo, Intergraph, etc.). Semantic spatial data integration poses great challenges to such systems. There has been a lot of work on spatial data integration and data exchange. Such work becomes the prerequisite for the construction of integrated, consistent spatial data warehouses.
Second, the realization of fast and flexible on-line analytical processing in a spatial data warehouse requires selectively materialization of spatial data cubes and fast online computation based on such partially materialized spatial data cubes [15]. Since multiple dimensions can be combined to form many combined spaces and each combined space may generate a good number of spatial maps with aggregated values, it is computationally challenging to precompute or materialize various aggregates with a huge number of spatial maps [10]. Various studies have been performed on partial materialization so that efficient computation and fast response can be achieved by exploring a limited number of precomputed subspace of a spatial data cube [16,17].

Key Applications

Spatial OLAP greatly facilitates online, multidimensional analysis and exploration of massive scale spatial data, and thus leads to its broad applications to those demanding multidimensional spatial data analysis. For example, for city administration, one can build concept hierarchies on multiple dimensions, such as time, location, population, and traffic, so that the traffic situation in the city can be checked by drilling and dicing to different districts and roads.

Future Directions

It is expected that an increasingly growing number of application-oriented spatial data warehouses and spatial data cubes will be constructed for massive-scale data exploration and analysis [12]. Also, sophisticated statical analysis tools will be integrated with spatial OLAP mechanisms for advanced spatial statistical analysis [14]. Moreover, it is expected that spatial data cube model will be extended to support advanced multidimensional data mining [6], as that has been done in prediction cubes [3]. Spatial OLAP is expected to be an important component in spatial data warehouse and spatial data mining systems [8].

Cross References

► Aggregation Query, Spatial
► Data Analysis, Spatial
► Data Warehouses and GIS
► Exploratory Visualization
► Generalization, On-the-Fly
► Metadata and Interoperability, Geospatial
► Statistical Descriptions of Spatial Patterns

Recommended Reading

1. Bédard, Y., Merrett, T., Han, J.: Fundamentals of geospatial data warehousing for geographic knowledge discovery. In: Miller, H.J., Han, J. (eds.) Geographic Data Mining and Discovery, pp. 53–73. Taylor and Francis, London, UK (2001)
2. Chaudhuri, S., Dayal, U.: An overview of data warehousing and OLAP technology. SIGMOD Record **26**, 65–74 (1997)
3. Chen, B.-C., Chen, L., Lin, Y., Ramakrishnan, R.: Prediction cubes. In: Proc. 2005 Int. Conf. Very Large Data Bases (VLDB'05), pp. 982–993. Trondheim, Norway, Aug. (2005)
4. Garcia-Molina, H., Ullman, J.D., Widom, J.: Database Systems: The Complete Book. Prentice Hall, Upper Saddle River, NJ (2002)
5. Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., Pirahesh, H.: Data cube: A relational aggregation operator generalizing group-by, cross-tab and subtotals. Data Min. Knowl. Discov. **1**, 29–54 (1997)
6. Han, J., Kamber, M.: Data Mining: Concepts and Techniques (2nd ed.). Morgan Kaufmann, London, UK (2006)
7. Harinarayan, V., Rajaraman, A., Ullman, J.D.: Implementing data cubes efficiently. In: Proc. 1996 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'96), pp. 205–216. Montreal, Canada, June (1996)
8. Miller, H., Han, J.: Geographic Data Mining and Knowledge Discovery. Taylor and Francis, London, UK (2001)
9. Papadias, D., Kalnis, P., Zhang, J., Tao, Y.: Efficient OLAP operations in spatial data warehouses. In: Proc. 2001 Int. Symp. Spatial and Temporal Databases (SSTD'01), pp. 443–459. Redondo Beach, CA, July (2001)
10. Pedersen, T.B., Tryfona, N.: Pre-aggregation in spatial data warehousess. In: Proc. 2001 Int. Symp. Spatial and Temporal Databases (SSTD'01), pp. 460–480. Redondo Beach, CA, July (2001)

11. Rigaux, P., Scholl, M.O., Voisard, A.: Spatial Databases: With Application to GIS. Morgan Kaufman, London, UK (2001)
12. Sarawagi, S., Agrawal, R., Megiddo, N.: Discovery-driven exploration of OLAP data cubes. In: Proc. Int. Conf. of Extending Database Technology (EDBT'98), pp. 168–182. Valencia, Spain, Mar. (1998)
13. Shekhar, S., Chawla, S.: Spatial Databases: A Tour. Prentice Hall, Upper Saddle River, NJ (2003)
14. Shoshani, A.: OLAP and statistical databases: Similarities and differences. In: Proc. 16th ACM Symp. Principles of Database Systems, pp. 185–196. Tucson, AZ, May (1997)
15. Stefanovic, N., Han, J., Koperski, K.: Object-based selective materialization for efficient implementation of spatial data cubes. IEEE Trans. Knowl. Data Eng. **12**, 938–958 (2000)
16. Zhang, L., Li, Y., Rao, F., Yu, X., Chen, Y., Liu, D.: An approach to enabling spatial OLAP by aggregating on spatial hierarchy. In: Proc. 2003 Int. Conf. Data Warehousing and Knowledge Discovery (DaWaK'03), pp. 35–44. Prague, Czech Rep., Sept. (2003)
17. Zhou, X., Truffet, D., Han, J.: Efficient polygon amalgamation methods for spatial OLAP and spatial data mining. In: Proc. 1999 Int. Symp. Large Spatial Databases (SSD'99), pp. 167–187. Hong Kong, China, July (1999)

One-Way-Out Evacuation

► Contraflow for Evacuation Traffic Management

On-Line Aggregation

► Aggregate Queries, Progressive Approximate

Online Generalization

► Generalization, On-the-Fly

Ontologies

► Geospatial Semantic Web

Ontology

► Geospatial Semantic Integration

Ontology, Spatio-temporal

► Temporal GIS and Applications

Ontology-Based Geospatial Data Integration

FREDERICO FONSECA
College of Information Sciences and Technology, The Pennsylvania State University, University Park, PA, USA

Synonyms

Information integration; Semantic information integration; Heterogeneity; Conflation; Database integration

Definition

Information integration is the combination of different types of information in a framework so that it can be queried, retrieved, and manipulated. This integration is usually done through an interface that acts as the integrator of information originating from different places. For integration to be efficient and to deliver the kind of information that the user is expecting, it is necessary to have an agreement on the meaning of the information. In a broader scope, it is necessary to reach an agreement about the meaning of the entities of the geographic world.

In order for information sharing to happen among different communities in a effective and meaningful way some preconditions are necessary. The concepts that people have about the real world must be explicitly formalized; such an explicit formalization of mental models is called an ontology. Ontology is often seen as an engineering artifact that describes a certain reality with a specific vocabulary, using a set of assumptions regarding the intended meaning of the vocabulary words. In philosophy ontology has a different meaning. For philosophy, ontology is the science that studies what exists. The fact that philosophers have been studying ontology since Aristotle may mean that their field can help information scientists learn to build good ontologies, but philosophy itself has different branches that have different assumptions about the world and how it is understood.

Historical Background

The effective integration of multiple resources and domains is known as interoperation. Interoperability is formally defined by the Open GIS Consortium as the "capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units." Efforts towards geographic information systems (GIS) interoperation are well documented. In the past, exchanging geographic information was as simple as sending paper maps or raw data tapes through the mail. Today, computers throughout the world are connected and the use of GIS has become widespread. The scope of interoperability has changed from static data exchange using flat files to global systems, interconnected using sophisticated protocols to exchange information on-line. In the future, computers are expected to be able to share not only information but also knowledge. Although GIS have been characterized as an integration tool, GIS interoperability is far from being fully operational.

Research on the integration of databases can be traced back to the mid 1980s, and today it is widespread among the GIS community. The complexity and richness of geographic information and the difficulty of its modeling raise specific issues for geographic information integration, such as the integration of different models of geographic entities (i. e., objects and fields) and different computer representations of these entities (i. e., raster and vector).

In GIS, the focus is changing from format integration to semantic interoperability. The first attempts to obtain GIS interoperability involved the direct translation of geographic data from one vendor format into another. A variation of this practice is the use of a standard file format. These formats can lead to information loss, as is often the case with the popular CAD-based (Computer Aided Design) format DXF (Drawing Exchange Format). Alternatives that avoid this problem are usually more complex, such as the Spatial Data Transfer Standard (SDTS) and the Spatial Archive and Interchange Format (SAIF).

The literature shows many proposals for the integration of information, ranging from federated databases with schema integration to ontologies. The new generation of information systems needs to handle semantic heterogeneity in making use of the amount of information available with the arrival of the internet and distributed computing. The support and use of multiple ontologies should be a basic feature of modern information systems if they are to support semantics in the integration of information. Ontologies can capture the semantics of information, can also be represented in a formal language, and can be used to store the related metadata, thus enabling a semantic approach to information integration. Sophisticated structures, such as ontologies, are good candidates for abstracting and modeling geographic information with the final objective of information integration.

This new generation of systems is characterized by the use of multiple ontologies and contexts to achieve semantic interoperability. Since Aristotle's theory of substances (objects, things, and persons) and accidents (qualities, events, and processes), ontology has been used as the foundation for theories and models of the world. Since ontology was first introduced, current research on ontology use can be found throughout the computer science community in areas such as computational linguistics and database theory. The areas that are being researched range from knowledge engineering, information integration, and object-oriented analysis to applications in medicine, mechanical engineering, and geographic information systems. The use of explicit ontologies contributes to the improvement of GIS. Since every information system is based on an implicit ontology, when the ontology is made explicit, conflicts are avoided between the common-sense ontology of the user and the mathematical concepts in the software, and conflicts between the ontological concepts and the implementation.

Scientific Fundamentals

According to Guarino "an ontology refers to an engineering artifact, constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary words. This set of assumptions has usually the form of a first-order logical theory, where vocabulary words appear as unary or binary predicate names, respectively called concepts and relations. In the simplest case, an ontology describes a hierarchy of concepts related by subsumption relationships; in more sophisticated cases, suitable axioms are added in order to express other relationships between concepts and to constrain their intended interpretation."

The purpose of ontologies is to bring more meaning (semantics) to the way data is collected, stored, and integrated. According to Guarino, an explicit ontology plays a central role in an ODIS (Ontology-Driven Information Systems) and drives all aspects and components of the system. Ontologies can be used at development time or at run time. In the geographic field, Fonseca proposed the use of ontologies in the development and use of GIS to enhance the integration of geographic information. He called those systems ontology-driven GIS.

Ontologies are usually created with one of two purposes. The first is to facilitate information integration, and the second is to facilitate communication between software agents. Both problems require what is usually called ontology integration, the mapping of concepts from one ontology to another, or as some approaches suggest, the creation of a common ontology from previously independent ontologies. From the technological point of view, for agents to be able to carry on a successful negotiation it is necessary that not only the ontologies be formally expressed, but also that they be expressed in a computer-readable language. Therefore, ontology languages are necessary.

The idea of ontology integration leads to great challenges. A fundamental barrier in the way of the development of fully general and reusable ontologies is what is called the

Tower of Babel problem. The difficulty is that insofar as database engineers attempt to accommodate, with the same database, groups of users possessing distinct ontologies (in the sense of preconceived theories), they must address the problem of integrating information in ways that are compatible with the perspectives of all significant potential users. This is a problem. It might be possible to integrate a limited number of alternative ontologies working out correspondences among them for a limited domain of data on a case-by-case, ad-hoc basis. However, such solutions are, by their nature, incompatible with the technological imperative behind the development of ontologies. They will be idiosyncratic, and are not general and reusable.

Accordingly, in order to achieve more general and reusable solutions, the use of techniques of logical and analytic philosophy to develop formal ontological structures with terminological consistency and subject to certain computationally convenient and efficient organizational principles was suggested. The problem of integrating databases derived from distinct ontologies is to be solved by requiring designers to conform, from the beginning, to an ontology. That is, the Tower of Babel problem is resolved by eliminating ontological differences at the outset, requiring all database designers to submit to first-order logic and/or whatever other formal and substantive constraints are compatible with a consistent ontology. Consequently, the complexity, subtlety and possibly surprising multidimensionality of the data, and the categories that organize them, must be limited in order to fit the needs of the database engineers. This kind of solution to the Tower of Babel problem is called the Newspeak Solution after George Orwell's introduction of the term in his novel Nineteen Eighty-Four. In order to meet the demands of the technological society envisaged by Orwell, there was a continual effort to create a reformed English, Newspeak, which was simpler, and less capable of expressing the ambiguity inherent in different points of view than traditional English. The consequence was that it became less expressive, and thus reduced the complexity of thought of those using it.

The difficulties associated with constructing a more complex alternative to Newspeak ontologies on a general scale are overwhelming to say the least. How, for example, could one provide a common or neutral framework for organizing and integrating all of the distinct descriptions that have been offered for any reasonably complex conceptual realm? The answer is, of course, that one cannot provide such a common ontology. If there is something like a common framework, it does not lie at the level of computational ontologies at all, but at the level at which users from different communities (paradigms) may learn to communicate with one another.

The main question is what is information used for? Formal approaches to ontology integration have to incorporate a hermeneutic dimension to information integration and interoperability issues. The three aspects of the conventions that structure human knowledge, analysis, synthesis, and application, are precisely the dimensions that are central to hermeneutics. A hermeneutic contextualization of ontology creation and integration can make room for communication among users who hold different points of view. Representation of diverse ontologies can be a setting within which users with differing conceptualizations of the world can learn to understand each other. Staying strictly within the ontological level of analysis, the problem of full-fledged information integration is insuperable. It is possible, however, to design a hermeneutic context: a place where users may come to learn from one another in a way much more fundamental than merely exchanging information within a mutually accepted paradigm. In order to do this, however, it is necessary first to explicitly recognize the hermeneutic context that is always present, though largely invisible when there are no disagreements about ontologies. For it is in this context that the adjudication of disagreements must go on. The key is to see that a database, as well as the world to which it refers, is itself an object of interpretation, and that, as such, those who use it are engaging in hermeneutic activity. Moreover, this activity of interpretation is strongly constrained by the applications users have in view. The use of hermeneutics in information integration provides a context from which it is possible to address the various problems facing ontologists and users: choice of ontological categories, ontology integration, and communication among users coming from different perspectives.

From such a perspective, information is seen as a process that is dependent on a certain preknowledge which the user of the data brings with him/her. Thus, the concept of the preunderstanding of a user of information and its extension to the preunderstanding of a community (and its ontology) is very close to the central role of presuppositions, or prejudices, in framing and guiding the emergence of experience in the work philosophical hermeneutics. Hence, attempts to develop frameworks aiming at information integration that will satisfy both the formalism and the practical and intuitive issues will have to deal with a hermeneutic approach.

Key Applications

Ontology Integration to Share Geographic Data About the Environment

Environmental researchers need improved integration tools and methods for a global sharing of scientific information. Historically, humans all over the world in every civilization have gathered information on the environment for a utilitarian purpose. This information is required to develop effective strategies for the conservation of the environment and design better policies to address common concerns about the environment. Environmental phenomena do not respect borders or frontiers, but the information resources, local knowledge and strategies used to combat them are all constructed independently and are not shared or directly sharable. Unfortunately, although sufficient information may exist to solve a particular problem, the existing data and metadata are inaccessible or not readily usable because they have been collected by different agents with a diversity of purposes. The type and quality of these data are also greatly influenced by the culture and language of the investigators. Thus, in order to be useful to other scientists in the world these diverse purposes need to be made explicit, with differences in meaning resolved, so that data can be understood by those among whom it is shared. However, the global character of environmental issues is a source of many impediments to the synthesis and utility of global data about the Earth. The three main factors are (1) language, (2) semantics and (3) culture, because although the researchers are studying the same subject they use different ontologies when collecting their data.

Future Directions

The use of semantic tools such as ontologies still has a long way to go. It is necessary to make the current technology easier to use. Meaning is ultimately established by the interpretation at the user level, and the tools available today are still too user unfriendly to enable a more effective use. Another challenge is the availability of good ontologies. Although there are some ontologies available, it is currently difficult to search and understand them, and thus difficult to effectively evaluate their usefulness.

Cross References

► Geospatial Semantic Integration
► Geospatial Semantic Web

Recommended Reading

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web: a new form of web content that is meaningful to computers will unleash a revolution of new possibilities. Sci. Am. **284**, 34–43 (2001)
2. Egenhofer, M.J.: Toward the semantic geospatial web. In: Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems, McLean, Virginia, 8–9 November 2002
3. Fonseca, F., Egenhofer, M., Agouris, P., Câmara, G.: Using ontologies for integrated geographic information systems. Trans. GIS **6**, 231–257 (2002)
4. Goodchild, M.: Geographical data modeling. Comput. Geosci. **18**, 401–408 (2002)
5. OpenGIS: The OpenGIS Guide—Introduction to Interoperable Geoprocessing and the OpenGIS Specification. Open GIS Consortium, Wayland, MA (1996)
6. Smith, B., Mark, D.: Ontology and geographic kinds. In: Proceedings of the 8th International Symposium on Spatial Data Handling, Vancouver, BC, Canada, 12–15 July 1998

Open Geospatial Consortium

MARKUS LUPP
lat/lon GmbH, Bonn, Germany

Synonyms

OGC; Open GIS Consortium

Definition

The Open Geospatial Consortium (OGC) is a nonprofit, international, voluntary consensus standards organization that is leading the development of standards for geospatial and location-based services.

Main Text

OGC is an international industry consortium of more than 300 companies, government agencies and universities participating in a consensus process to develop publicly available interface specifications. OpenGIS Specifications support interoperable solutions that "geo-enable" the web, wireless and location-based services, and mainstream IT. The specifications empower technology developers to make complex spatial information and services accessible and useful with all kinds of applications.

OpenGIS is a registered trademark of the OGC and is the brand name associated with the specifications and documents produced by the OGC. OpenGIS specifications are developed in a unique consensus process supported by OGC industry, government and academic members to enable geoprocessing technologies to interoperate, or "plug and play". You will also find the OpenGIS trademark associated with products that implement or comply to the OGC specifications. Make sure that your geoprocessing and location services procurement and technology development programs demand OpenGIS specifications!

Cross References

► deegree Free Software
► Geography Markup Language (GML)

References

http://www.opengeospatial.org

Open GIS Consortium

► Open Geospatial Consortium

Open Source

► deegree Free Software
► Geography Markup Language (GML)
► PostGIS
► Quantum GIS

Open Standard

► Scalable Vector Graphics (SVG)

Open Standards

► OGC's Open Standards for Geospatial Interoperability

Open-Source GIS

► Environmental Modeling Using Open Source Tools
► MapWindow GIS

Open-Source GIS Libraries

ANDREAS NEUMANN, MARCO HUGENTOBLER
Institute of Cartography, ETH Zurich, Zurich, Switzerland

Synonyms

GDAL; OGR; PROJ; Java Topology Suite (JTS); Java Conflation Suite (JCS); Geometric Engine Open Source (GEOS); GPSBabel ; OGIS; Gnu Public License (GPL)

Definition

Open source GIS libraries provide basic functionality for certain aspects and tasks of open source and commercial GIS software. Libraries are helper software which offers services to independent GIS software, thus enabling code and data to be shared in a modular way. If a software provides an additional abstraction layer, underlying libraries can be exchanged if needed, e. g., due to performance, accuracy or functionality reasons. In GIS, some basic functionality is required by many derived and specialized GIS software and tools. Rather than implementing the same functionality again and again, specialized libraries can provide the functionality, offering derived software a head start rather than having to write everything from scratch. Typical cases where libraries are traditionally used are graphics and GIS format support and conversion, reprojection support, computational geometry operations, topology operations, and more. The goal of this entry is not to provide a complete list of available open source GIS libraries, which would be quickly outdated, but to describe the availability and functionality of some of the more popular GIS libraries. *GDAL/OGR* is a GIS and image format access and conversion library and a suite of utilities (GDAL is responsible for raster data and OGR for vector data access). *PROJ* is a reprojection library. The *Java Topology Suite (JTS) and GEOS* both provide geometry engines for computational geometry and topologic queries. The *Java Conflation Suite (JCS)* provides functionality and tools for combining, integrating and improving data from various data sources. *GPSBabel* enables the reading, writing and conversion of various GPS formats. Many open source GIS libraries are published under a less strict license, such as a variation of the MIT License or the LGPL, thus also allowing commercial use of the libraries without forcing a company to release its full source code of depending applications.

Historical Background

Naturally, since multiple projects are discussed, each project has its own history.

The history of GDAL/OGR dates back to 1993, when Frank Wamerdam (the main programmer of GDAL/OGR) started to work at PCI Geomatics [7]. Wamerdam initially worked at PCI on a predecessor library of GDAL/OGR called GDB (Generic Database). In 1998 he became an independent consulter in Open Source GIS, filling the empty niche for a library to read, write, convert and manipulate geospatial raster and vector data. GDAL is an abbreviation for *Geospatial Data Abstraction Library*. The name of OGR has historic roots (*OpenGIS Simple Features Reference Implementation*), though it has lost its meaning today since it is no longer a reference implementation. In 2006, GDAL/OGR joined the OSGeo (Open Source Geospatial) foundation as a founding member and began transitioning to a more community oriented governance model. Today, GDAL/OGR is used by most open source and a lot of commercial GIS software. While there are some additional contributors, Wamerdam is still the primary developer of the library. Currently, GDAL supports more than 60 raster formats and OGR supports more than 25 vector formats and spatial database providers. However, not every operation is supported by each format driver.

The history of *PROJ.4* [1] started in the late 70's, when Gerald Evenden was involved in the development of map plotting software at the Atlantic Geology branch of USGS.

At that time, two separate software packages had been developed: MAPGEN for map plotting and PROJ for projection calculations. This separation was a wise decision because projection calculations are useful in many work flows, not just for making maps. Additionally, at that time, other available projection libraries were either not up to the task or impossible to integrate into other software. In the 80's, the software packages went through several iterations with different programming languages until ending up as a C library. With the advent of commercial graphic software, MAPGEN was abandoned, while PROJ was further developed. Originally designed for US datums and projection systems, the new library called *libproj4* concentrates on projections, handles international projections and does not deal with datum conversions. Libproj4 is a new development and not directly compatible with the old PROJ system and the *PROJ.4* distribution which is more widely used today. PROJ.4 is now maintained by Frank Wamerdam, but still mostly based on Gerald Evenden's work.

The first version of the *Java Topology Suite* (JTS) was released in February 2002, developed by Vivid Solutions and sponsored by several Canadian and British Columbian governmental institutions. As the name already reveals, the software is written in Java. Updates occurred at regular intervals, enhancing functionality, performance and providing bug fixes. The latest release, 1.8, dates from December 2006. The Java Conflation Suite builds on the Java Topology Suite and ties into the JUMP user interface. Its first release dates from June 2003 with a compatibility release in November 2003 to collaborate with the new version of JUMP. GEOS (*Geometry Engine Open Source*) is a C++ port of the Java Topology Suite. Release 1.0 dates from November 2003 and the latest release is from February 2007. GEOS is developed by Refractions Research, the same company responsible for developing PostGIS and other Open Source GIS tools.

Version 1.0 of *GPSBabel* was released by Robert Lipe in October 2002. The first Windows GUI release dates from December 2002 and an OSX GUI was contributed in February 2004. The latest release, 1.3.2, is from November 2006. GPSBabel currently supports about one hundred different GPS dialects and versions and is available for almost any platform.

Scientific Fundamentals

GDAL

GDAL is a library to access raster data formats and presents a single abstraction model for all the formats to an application. GDAL is written in C++, but also offers bindings for C and Python. GDAL is licensed with an MIT style open source license. This license basically excludes any warranty and states that everything can be done with the software (apart from removing the copyright disclaimer) [2].

The GDAL abstraction model is based on the representation of a dataset (GDALDataset) and the representation of a raster band (GDALRasterBand). GDALDataset contains information about the georeferencing of the raster and references to the raster bands. GDALRasterBand provides the method GDALRasterBand::GDALRasterIO to access data. This method loads the raster values to a memory buffer where they can be accessed by the program using GDAL. Two important parameters of this method are the resolutions in x- and y-direction. Note that, used with different arguments, the same method can also be used for writing data.

A raster band can have associated overviews (also called pyramids). In fact, overviews are also GDALRasterBands, but with a lower resolution than the original band. This makes queries with lower resolutions much faster. Overviews can be created for all bands in a dataset with the method GDALDataset::createOverviews, giving a list with decimation factors for the requested overviews.

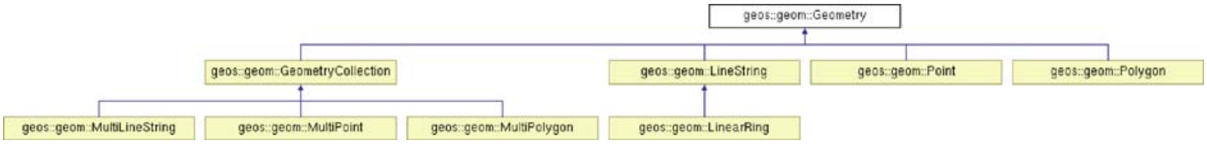
OGR

The code of the OGR library is not dependent on GDAL. However, it resides in the same source tree and is compiled into the same binary library (libgdal). OGR supports access to a large number of vector data formats, including GML, ESRI shapefiles, GRASS and PostgreSQL [5]. Sequential reading of the data is provided for all supported formats. Further operations may be available depending on the data formats and if the OGR drivers are able to return their capabilities at runtime. Such capabilities include random read, sequential write, random write, delete feature, fast spatial filter, fast feature count, fast query of layer extent. OGR is also able to create new empty data sources for some formats.

The most important objects in OGR are the layers (OGRLayer), the features (OGRFeature) and the geometries (OGRGeometry). A sequential read of the features in a layer can be done by using the functions OGRLayer::ResetReading() and then repeatedly OGRLayer::GetNextFeature(). OGRFeature contains the attribute values and a reference to the feature geometry. OGRGeometry is an abstract base class, implemented by specific subclasses for the representation of point, multipoint, line, multiline, polygon, multipolygon.

Java Topology Suite (JTS)

The Java Topology Suite (JTS) is a library (API) for 2D spatial functions and predicates. It implements the spatial



Open-Source GIS Libraries, Figure 1 Geometry Classes in GEOS, also apply to JTS (with different namespaces); Source: [3]

functions and predicates defined in the OGC Simple Features for SQL specification. The JTS implementation contains complete, consistent and robust implementation of fundamental spatial algorithms that is fast enough for production use. As the name indicates, it is written in Java and is released under the LGPL (Lesser GNU Public License). JTS builds on the spatial data types defined in the OGC simple features for SQL specification and supports Points, MultiPoints, LineStrings, MultiLineStrings, Polygons, MultiPolygons, LinearRings and GeometryCollections (see Fig. 1). JTS supports the following spatial analysis methods:

- Intersection
- Union
- Difference
- Symmetric Difference
- Convex Hull
- Buffer (positive and negative offsets)

Binary predicate methods take two geometries as arguments and return a boolean result indicating whether the given spatial relationship exists or not. JTS supports the following binary spatial predicates:

- Equals
- Disjoint
- Intersects
- Touches
- Crosses
- Within
- Contains
- Overlaps

Additionally, the *relate* operator is supported which works according to the *Dimensionally Extended 9 Intersection Matrix* (DE-9IM).

GEOS

GEOS is a C++ port of the Java Topology Suite (JTS) and also offers bindings C and python. Even if GEOS itself is written in C++, it is recommended to use the C interface as it is supposed to remain stable, while the C++ interface may differ between versions [3].

A geometric feature is represented in GEOS by the abstract class geos::Geometry, from which representations for the simple features are derived (see Fig. 1). The functionality of geos::Geometry includes the calculation of convex hulls, centroids and buffers as well as the intersection, union and difference of two features. A geos::Geometry object may be created from vertex coordinates by using the methods of geos::GeometryFactory. GEOS also provides the possibility to import geometries from Well Known Text (WKT) and Well Known Binary (WKB) format.

Java Conflation Suite (JCS)

The Java Conflation Suite (JCS) [8] is an API/library and a set of interactive tools assisting while conflating (merging) and cleaning vector data sets. It is part of the JUMP GIS framework and builds upon the Java Topology Suite. It was released under the terms of the GPL (Gnu Public License). During conflation, it helps with coverage cleaning, coverage alignment and road network matching. JCS also assists in quality assurance of vector data sets by detecting and visualizing errors and offering both automatic and manual cleaning functions. All detection and cleaning mechanisms are accessible either programmatically or interactively.

The coverage cleaning and conflation functions include the following:

- detect and clean gaps
- detect and clean overlaps
- snap vertices
- detect and fix boundary alignment errors (horizontal conflation)
- detect and fix coverage alignment errors (vertical conflation), one data set is the reference data set
- road network matching: JCS provides algorithms to establish node and edge matchings between two road networks
- precision reduction: reduce excessive precision to lower precision. This step may produce incorrect vector topology, which can be fixed in an extra step (see above)
- geometry difference detection: find geometric differences of two versions of the same data set. JCS supports either exact matching or matching with a distance tolerance

GPSBabel

GPSBabel [4] helps to *flatten the tower of babel* that GPS device and software manufacturers introduced with their proprietary GPS data formats. Despite the fact that standard GPS data formats (like GPX) exist and GPS data isn't really of complex nature, there is still a considerable amount of GPS data stored and exchanged with proprietary and incompatible data formats. GPSBabel can translate waypoints, routes, tracks and partially signal quality (PDOP) information between over a hundred different data formats and versions. Additionally, it can filter, interpolate and rearrange (sort) data and batch process many files in one run. Strictly spoken, GPSBabel isn't a library but a commandline tool. The fact that it runs as a commandline tool helps create workflows where GPSBabel is part of a workflow pipeline controlled by a shell script, batch file or any programming language and makes it suitable for web server usage. However, there are also GUI tools available for the occasional user. As most open source software, GPSBabel works on almost any platform (including Linux/Unix, MacOSX and Windows). GPSBabel is also used by Google Earth and the geocaching community. It is licensed under the Gnu Public License (GPL).

GPSBabel supports powerful filter options. It may detect and remove duplicate coordinates, reduce the number of points by using a simplify operator, filter data by geometry (within the radius of a given point, inside a given polygon, within the distance of a given arc). The simplify option is particularly useful for devices with limited storage capabilities. The algorithm tries to preserve the original geometry as well as it can, honoring the specified parameters. A PDOP filter allows one to filter out unreliable points with a bad PDOP value. This filter works separately for HDOP (horizontal precision) and VDOP (vertical precision) values. The rearranging options allow sorting by time, shortname, name, description and geocache id. The interpolation functions allow one to add points at regular time and distance intervals. Finally, there are conversion and manipulation functions, such as shifting all timestamps, splitting and merging tracks and files, convert waypoints to tracks and convert tracks to waypoints.

Key Applications

Base Functionality for Derived GIS Software

The functionality of the libraries can be used to build a GIS on top of it. With this approach, a GIS program can be built by providing the a graphical user-interface to translate user requests to calls to library functions. An example of such a program is QGIS [6]. QGIS uses GDAL/OGR for reading and writing to data sources, PROJ4 for reprojecting vector layers on the fly and GEOS for intersection tests between geometries and (selection) rectangles.

Format Conversion, Import/Export, Data Access

GDAL and OGR provide, in addition to the library, command line tools for data access and format conversion. For instance, *ogrinfo* prints out information about vector data sources and *ogr2ogr* converts between vector formats. *gdaltranslate* is the tool to convert raster formats and *gdalwarp* reprojects raster datasets. *gdal_rasterize* converts vector files to raster files through rasterization, *rgb2pct* and *pct2rgb* allows conversion between 24bit raster files and paletted 8bit raster files.

Data Management and Administration

As mentioned in the previous paragraph, various commandline utilities exist to receive information on data sets, convert formats, and reproject data. *gdaladdo* is a utility to add overview pyramids with lower resolution to a high resolution raster data set. *gdaltindex* and *ogrtindex* create index files documenting the footprint of a raster data set for quick retrieval, which is in particular important for map servers. *gdal_merge* builds mosaics from different raster data sets. *gdal_contour* can create vector contours from a raster based digital terrain model. The Java Conflation Suite assists for merging vector data sets. GPSBabel helps to manage and process GPS data. The JTS and GEOS library can help to select, filter and manipulate data based on geometry or spatial relationships.

Merging of Vector Data Sets, Consistency Checks, Quality Assurance

As mentioned in the JCS section, the Java Conflation Suite can assist with the merging of vector (feature) data sets and helps to check data quality and consistency. It can detect and visualize errors and consistency problems and offers both automatic and manual (interactive) cleaning tools. It is accessible to developers through APIs or to end-users through a graphical user interface, utilizing the JUMP GIS framework.

Building Blocks for More Complex Work Flows and Web Applications

As many libraries also provide commandline based utilities, it is easy to write batch files, shell scripts or other software that utilizes the data management and administration capabilities of the libraries dealt with in this entry. They can also be used as components for web GIS and web cartography in combination with server side scripting or programming languages.

Future Directions

GDAL/OGR will provide more language bindings, featuring support of Java and .NET access to the libraries. Additionally, Wamerdam will work on a thread safe version of the libraries. Obviously, users will demand more drivers targeting data formats not yet supported. Unifying the architecture of GDAL and OGR is also on the agenda. Future versions of *PROJ.4* aim to provide better documentation, improved datum shifting support and better APIs for other software to build upon PROJ.4. The Java Topology Suite and its C++ port GEOS are, functionality wise, almost complete. Future releases will concentrate on bug fixes, performance and compatibility issues. The functionality of future versions of the Java Conflation Suite will depend on functionality ordered by customers or research projects. The future is thus hard to predict. It would be desirable that JCS would support additional topology rules. GPSBabel obviously will support upcoming GPS data formats and improve support for new visualization tools such as Google Earth and other software projects. As with any other open source project, the future is difficult to predict. New functionality will be added if there is a need to, a client pays for development or a programmer invests time and interest to implement it. Possible improvements will be ports to other platforms, improved GUIs or support for additional data formats.

Cross References

- ► ArcGIS: General Purpose GIS Software System
- ► Conflation of Features
- ► deegree Free Software
- ► GRASS
- ► PostGIS
- ► Pyramid Technique
- ► Quantum GIS
- ► Raster Data
- ► University of Minnesota (UMN) Map Server
- ► Vector Data

Recommended Reading

Evenden, G.: Brief History of PROJ[4] (2004) http://lists.maptools.org/pipermail/proj/2004-March/001117.html. Accessed 17 Feb 2007

GDAL: GDAL – Geospatial Data Abstraction Library (2007) http://www.gdal.org. Accessed 22 Feb 2007

GEOS (2007): GEOS homepage (2007) http://geos.refractions.net. Accessed 22 Feb 2007

GPSBabel Community (2007): GPSBabel: convert, upload, download data from GPS and Map programs (2007) http://www.gspbabel.org/. Accessed 20 Feb 2007

OGR (2007): OGR Vector Formats (2007) http://www.gdal.org/ogr/ogr_formats.html. Accessed 22 Feb 2007

QGIS Community (2007): QGIS Community – Home (2007) http://qgis.org. Accessed 22 Feb 2007

QGIS Community (2005): QGIS Community – Interview with Frank Warmerdam (2007) http://qgis.org/content/view/58/44/. Accessed 17 Feb 2007

Vivid Solutions (2006): Java Conflation Suite (JCS) (2007) http://www.vividsolutions.com/JCS/. Accessed 20 Feb 2007

Operation, Vague Spatial

► Vague Spatial Data Types

Operator, Metric

► Spatio-temporal Query Languages

Operator, Spatial

► Spatio-temporal Query Languages

Operator, Temporal

► Spatio-temporal Query Languages

Operator, Topological

► Spatio-temporal Query Languages

Optimal Prediction

► Kriging

Optimal Sequenced Route Query

► Trip Planning Queries in Road Network Databases

Optimum Interpolation

► Kriging

Oracle Geographic Information System

► Oracle Spatial, Geometries

Oracle Locator

► Oracle Spatial, Geometries

Oracle Spatial

- ► Oracle Spatial, Geometries
- ► Smallworld Software Suite

Oracle Spatial, Geometries

RAVI KOTHURI, SIVA RAVADA
Spatial Technologies, Oracle USA, Nashua, NH, USA

Synonyms

Oracle locator; Oracle spatial; Oracle geographic information system; Reference system, spatial; OGC, OGC simple features; SQL/MM; ISO/IEC; Minimum bounding rectangles; Indexing, spatial; Populating, topology

Definition

A geometry in Oracle Spatial can be one of the following[1]:

- Point
- Linestring connecting two or more points
- Polygon specified as a closed linestring and indicating an area bounded by the linestring (with zero or more holes inside the outer bounding linestring)
- Multipoint collection
- Multiline collection consisting of unconnected linestrings
- Multipolygon collection consisting of nonoverlapping polygons, or
- Heterogeneous collection consisting of a mixture of point, linestring, and/or polygon geometries

Each geometry can be specified in the context of a *spatial reference system*, also referred to as a *coordinate system*. Typical coordinate systems are *local* or *geographic*. Geographic coordinate systems are *geodetic* or *projected*. Whereas local and projected coordinate systems operate in an Euclidean space, geodetic coordinate systems operate on the Earth's surface.

Geometry data in Oracle Spatial can be stored using (1) the SDO_GEOMETRY data type or using (2) the SDO_TOPO_GEOMETRY data type. Whereas the SDO_GEOMETRY type stores the geometry as is, the SDO_TOPO_GEOMETRY stores it using simple topological primitive elements such as nodes, edges and faces. The latter representation allows for shared representation of edges (and other elements) between different feature geometries. For instance, a line (or edge) can be shared between the boundary of a land parcel as well as a road segment. The advantages of such shared representation are (1) the ease of topology management across multiple features: changes in the nodes, edges and faces automatically reflect in the higher-level geometry features; (2) elimination of redundancy translating into less storage space; and (3) sometimes faster processing for topological queries.

[1] Additional types to model geometric shapes in three-dimensional space are being added in subsequent releases of Oracle. This entry focuses only on two-dimensional geometries.

Historical Background

For several decades, defining, storing and manipulating geometry data has received much attention in various research communities including geographic information systems (GIS), computer-aided design (CAD)/ computer-aided manufacturing CAM, medical and virtual reality. Whereas the GIS industry deals with mapping points, lines and polygons on the surface of the earth using two-dimensional (2D) coordinates, the CAD/CAM industry deals with representing buildings, machines and other manmade artifacts usually in mostly three-dimensional (3D) spaces. The medical and virtual reality applications use a combination of 2D and 3D objects. The formats for representing the geometry data have varied across these fields, and also among different vendors in each of these fields. For instance, the spatial vendors such as Oracle Spatial, ESRI, Mapinfo, Autodesk, Bentley, and Intergraph each have their own formats to store 2D geometry data.

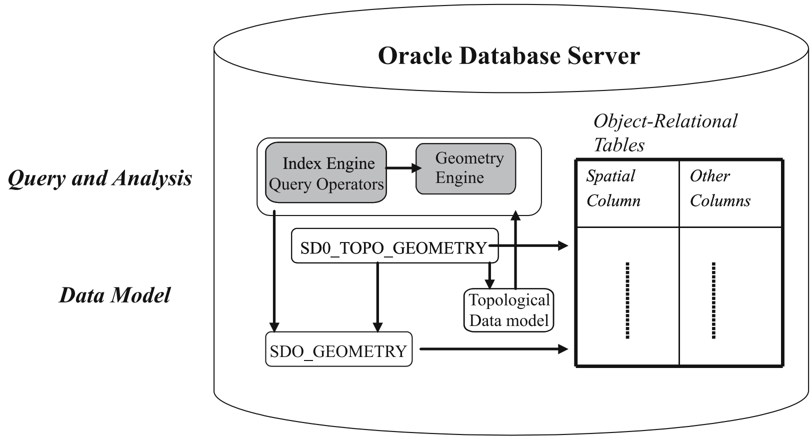
Over the past few decades, several commercial and research products for managing geometry data have emerged. The majority of the initial products either stored them in a file-based ASCII format or some vendor-specific proprietary binary formats (only understood by vendor tools). Storage in file systems has its limitations with regard to size, query/access, scalability, and recoverability. Storing in binary formats makes the data uninteroperable. To make the data across different vendors interoperable, the GIS community has defined two standards:

- The Open GIS Consortium (OGC) has the Simple Features specification[2] for storage, retrieval, query, and update of simple geospatial features. This specification defines a geometry type with appropriate subtypes to model points, line strings, polygons, and so on.
- Geometry data is explicitly dealt with in Part 3, of SQL/MM[3] (the ISO/IEC[4] international standard for "Text, Spatial, Still Images, and Data Mining"). This specification deals with spatial user-defined types and associated routines to store, manage, and retrieve spatial data. This standard specifies the ST_Geometry type to store spatial data. This type has subtypes such as ST_Point, ST_LineString, and ST_Polygon to mod-

[2] Open GIS Consortium, *OpenGIS Simple Features Specification for SQL Revision 1.1*, http://www.opengis.org/docs/99-049.pdf, May 5, 1999.

[3] Structured Query Language (Multimedia)

[4] International Organization for Standardization/International Electrotechnical Commission.



Oracle Spatial, Geometries, Figure 1 Geometries in Oracle Spatial

el different types of spatial geometries. This standard also specifies a well-known text format for specifying geometries. For instance, the string "POINT(1 1)" indicates a point geometry with coordinates at (1,1). Note that the types represented in OGC Simple Features are a subset of those defined by SQL/MM Part 3.

Oracle Spatial's model for storing geometry data is an open format, i. e., a nonbinary representation that can be read by any vendor. Furthermore, the data is stored as first-class objects inside an Oracle database. This extends all the salient features of a database such as atomicity, concurrency, scalability and recoverability to geometry data. Users can query spatial and nonspatial data in a single query. In addition to storing the geometry data as an open format (i. e., a format that can be read/understood by nonOracle software), Oracle Spatial is also working on defining web services on top of the open SDO_GEOMETRY type to interoperate with other GIS applications.

Scientific Fundamentals

Figure 1 shows the architecture for geometry storage and manipulation in Oracle Spatial. The data model has two main data types, SDO_GEOMETRY and SDO_TOPO_GEOMETRY, to store the geometry data. These types can be stored as columns of a table in Oracle DB. After storing geometry data in Oracle tables, users can operate on them using appropriate spatial query operators and functions. These operators are processed in a multistep refinement process: for instance, using a spatial index engine and then a geometry engine (see [1] for more details) when the data are stored as SDO_GEOMETRY objects.

Figure 2 shows a sample of typical geometry data in GIS applications: a polygon geometry representing the "city hall" property boundary and an adjoining street "City Hall St". An illustration of how to store and manipulate this data using the two alternative models for storing geometries in Oracle Spatial – the SDO_GEOMETRY and the SDO_TOPO_GEOMETRY follows.

The SDO_GEOMETRY Model

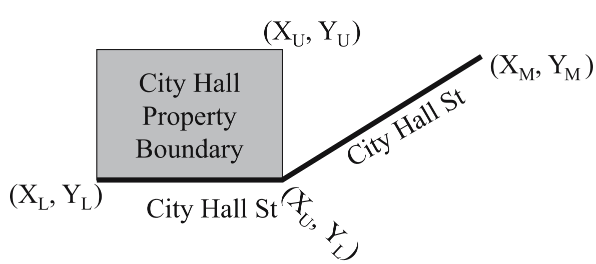
Storage To store properties and streets in the SDO_GEOMETRY model, spatial users can create tables of the form:

```
CREATE TABLE city_properties
(property_name VARCHAR2(32),
   property_boundary SDO_GEOMETRY);
CREATE TABLE city_streets
(street_name VARCHAR2(32),
   street_geometry SDO_GEOMETRY);
```

Geometry data can be inserted into these tables as follows:

```
INSERT INTO city_properties
('CITY HALL',
        SDO_GEOMETRY(2003, 8307, null,
        SDO_ELEM_INFO_ARRAY(1,1003,3),
        SDO_ORDINATES(XL, YL, XU, YU))
);
```

Example 1 SQL for inserting a geometry into the city_properties table



Oracle Spatial, Geometries, Figure 2 Typical geometry data

Observe that the second column instantiates an SDO_GEOMETRY object and inserts that into the property_boundary column of the city_properties table. The SDO_GEOMETRY object has the following fields (exact details in [2]):

- The SDO_GTYPE attribute specifies which type of shape (point, line, polygon, collection, multipoint, multiline, or multipolygon) that the geometry actually represents. In Example 1, the SDO_GTYPE is set to 2003, where the "2" indicates the dimensionality and the last digit "3" indicates it is a polygon geometry.
- The SDO_SRID attribute specifies the ID of the spatial reference system (coordinate system) in which the location/shape of the geometry is specified. In Example 1 this is set to 8307.
- If the geometry is a point (e. g., the location of customers), then users can store the coordinates in the SDO_POINT attribute of the SDO_GEOMETRY. Since the "city hall" geometry is not a point, this field is set to NULL.
- If the geometry is an arbitrary shape (e. g., a street network or city boundaries), then users can store the coordinates using the SDO_ORDINATES array and SDO_ELEM_INFO array attributes.
  - ORDINATES attribute stores the coordinates of all elements of the geometry.
  - The SDO_ELEM_INFO attribute specifies where in the SDO_ORDINATES array a new element starts, how it is connected (by straight lines or arcs), and whether it is a point (although the SDO_POINT is recommended for storage and performance reasons as explained below), a line, or a polygon. In the above example, the SDO_ELEM_INFO is set to (1,1003,3) indicating a rectangle and the lower vertex (X_L, Y_L) and upper vertex (X_U, Y_U) coordinates being specified in the SDO_ORDINATES array.

Other simple examples are point and line geometries. For instance, users can insert a a line geometry to represent a street adjoining the "city hall" as follows:

```
INSERT INTO city_streets
("CITY HALL ST",
        SDO_GEOMETRY(2002, 8307, null,
        SDO_ELEM_INFO_ARRAY(1,2,1),
        SDO_ORDINATE_ARRAY(X_L,Y_L,
                X_U,Y_L, X_M,Y_M))
);
```

Example 2 SQL for inserting a geometry into the city_streets table

Note that the geometries for both "CITY HALL" and "CITY HALL ST" share the edge <(X_L, Y_L), (X_U, Y_L)> and is stored in both of them. If for some reason, the street boundary is modified to go inward into the city hall property, both geometries need to be updated. The SDO_TOPO_GEOMETRY model, discussed below, eliminates the need for such redundant storage and multiple updates.

Note that if the geometry is just a point, the SDO_GTYPE will be 2001 and the coordinates are stored in the SDO_POINT attribute. More detailed examples (such as multipoints, multilines, multipolygons and collections) can be found in Table 4-1 of [2].

Spatial Functions After storing geometries in the city_properties table, users can perform a variety of spatial operations including:

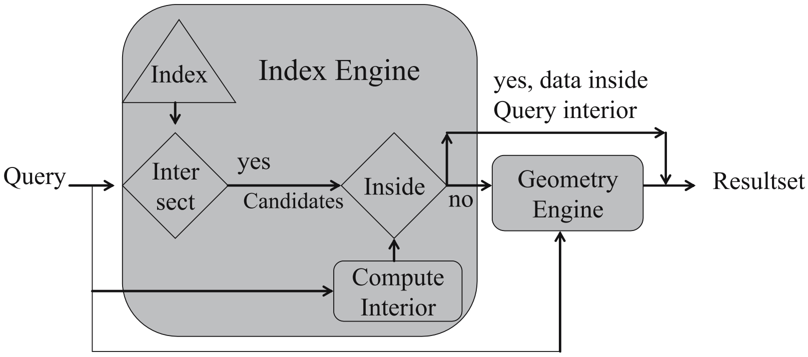
- SDO_GEOM.Relate: this function can determine the relationship of one geometry with another geometry. For instance, users can identify all city_properties that touch a query region r as follows:

```
SELECT * FROM city_properties
WHERE SDO_GEOM.RELATE
      (property_boundary, r, "TOUCH",
<tolerance_value>)="TRUE"
```

- SDO_GEOM.SDO_BUFFER: this function returns a buffer around a specified geometry. Users can use this function to determine the geometry that defines say, the 2-mile, radius around "city hall".
- SDO_GEOM.SDO_UNION, SDO_GEOM.intersection, ...: these functions allow users to compute the union, or intersection of a pair of geometries.

Spatial Indexing and Query Processing Identifying the properties that touch the specified query using the SDO_GEOM.RELATE function can be slow, especially when the city_properties table has a large number of rows. A better approach is to use an index to prune the search. Oracle Spatial uses a variant of R*-trees (see [3] for details) and several other optimizations [1,5,6] for indexing spatial data. The spatial index stores approximations such asminimum bounding rectangles (MBRs) for each geometry. Queries are then compared with the MBR approximations first, and only appropriate candidates are passed on to an exact geometry–geometry comparison in the geometry engine (using the SDO_GEOM.RELATE and other spatial functions). This multistep processing filter out a majority of irrelevant geometries. The architecture for query processing is shown in Fig. 3.

Oracle Spatial has introduced additional optimizations such as those described in [1]. For each query, an interior component is identified. Any data geometries that are completely inside the interior of the query are included in the



Oracle Spatial, Geometries, Figure 3 Query processing on SDO_GEOMETRY data in Oracle

result set, bypassing the expensive geometry-engine processing. It has been show that this optimization is quite effective for large query windows and obtains substantial query speed increases [1].

The SDO_TOPO_GEOMETRY Model

In Examples 1 and 2 (Fig. 2), observe that the edge <(X_L, Y_L), (X_U, Y_L)> is shared between two SDO_GEOMETRY objects but stored in both. Such redundant storage not only increases storage requirements, but also necessitates *explicit* maintenance of the integrity between multiple geometry tables. For consistency with Oracle documentation, the geometry tables are hereafter referred to as *feature tables* and geometries in these tables as *features*.
In the SDO_TOPO_GEOMETRY or simply "topology" model, the city_properties and city_streets are created as follows:

```
Create Table City_Properties (
Feature_Name VARCHAR2(30) PRIMARY KEY,
Feature SDO_TOPO_GEOMETRY);
```

```
Create Table City_Streets (
Feature_Name VARCHAR2(30) PRIMARY KEY,
Feature SDO_TOPO_GEOMETRY );
```

Where the SDO_TOPO_GEOMETRY has the following structure:

```
TG_Type                 NUMBER
TG_ID                   NUMBER
TG_Layer_ID             NUMBER
Topology_ID             NUMBER
```

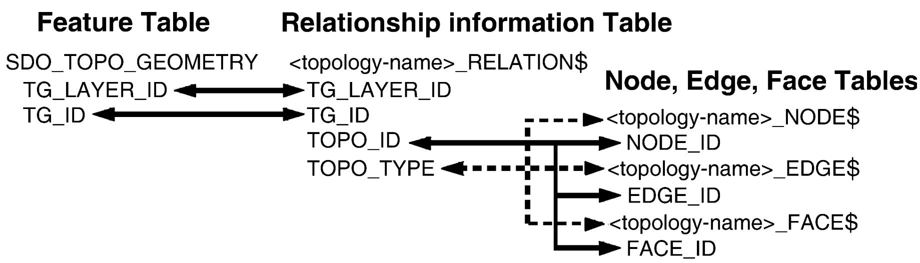
Together the set of feature layers constitute a "TOPOLOGY". For instance, users can create an explicit topology called "CITY MODEL" and add the feature layers "CITY_PROPERTIES" and "CITY_STREETS" to it.
When a TOPOLOGY is created, three dependent tables, <TOPOLOGY>_NODE$, <TOPOLOGY>_EDGE$, <TOPOLOGY>_FACE$, (or NODE$, EDGE$, and FACE$ for short) are created by Oracle Spatial. These tables respectively store the node-, edge- and face-type primitive elements of the specified topology.
Each SDO_TOPO_GEOMETRY maps to primitive elements (that compose the feature) in the NODE$, EDGE$, and FACE$ table. This mapping is maintained in another system-generated RELATION$ table of the following structure:

```
TG_Layer_ID             NUMBER
TG_ID                   NUMBER
Topo_ID                 NUMBER
Topo_type               NUMBER
```

Note that the <TOPO_ID, TOPO_TYPE> refer to the ID, and the type (node, edge, face) of the primitive element in the feature. This is described in more detail in Fig. 4.

Populating a TOPOLOGY A simple way to populate a topology is by populating the NODE$, EDGE$, and FACE$ tables for primitive elements directly. For instance, the EDGE$ table could store the edge from (X_L, Y_L), to (X_U, Y_L) as TOPO_ID=1, TOPO_TYPE=2. This edge can be referred to during the construction of features as follows:

```
INSERT INTO City_Streets VALUES
("City Hal St",
   SDO_TOPO_GEOMETRY(
      <TOPOLOGYname>,
   2002, - LINE FEATURE
      2, - TOPOLOGY_FEATURE_LAYER ID
      SDO_TOPO_OBJECT_ARRAY(
      - Array of lowerlevel primitives
      SDO_TOPO_OBJECT(1, 2)
      - Refer to Lower-level primitive
      - i.e., the edge
      - <(X_L, Y_L),(X_U, Y_L)>
        SDO_TOPO_OBJECT (2,2)
                        - another edge
        )
      );
```



Oracle Spatial, Geometries, Figure 4 Relationship between SDO_TOPO_GEOMETRY feature and the corresponding primitives

Likewise, the features in the CITY_PROPERTIES layer can be constructed.

An alternative approach is to insert an SDO_GEOMETRY as a feature into a topology. The following SQL shows an example of how to insert the linestring SDO_GEOMETRY connecting (X_L, Y_L), (X_U, Y_L,) and (X_M, Y_M) for the "CITY HALL ST" feature directly into the City_Streets feature layer.

```
INSERT INTO City_Streets
   VALUES ("City Hal St",
      sdo_topo_map.create_feature
                        ("<TOPOLOGYname>,
   SDO_GEOMETRY(2002,null,null,
          sdo_elem_info_array(1,2,1),
   SDO_ORDINATE_ARRAY
          (XL, YL, XU, YL, Xm, Ym) );
```

Indexing and Querying on Features Observe that the SDO_TOPO_GEOMETRY object in the feature tables does not store the "geometry" explicitly but only *points* (through its feature_id) to the topological primitives in the RELATION$ table. If the exact geometry needs to be materialized, the GETGEOMETRY method in the SDO_TOPO_GEOMETRY type can be utilized. For instance, the following SQL returns the geometry as an SDO_GEOMETRY from the CITY_STREETS feature layer.

```
SELECT feature.GETGEOMETRY() FROM
city_streets;
```

Note that the above function is evaluated by obtaining the composing primitive elements for the SDO_TOPO_GEOMETRY. This will be slower than storing the entire geometry as is as in the SDO_GEOMETRY model.

In most queries, users are more interested in identifying features that intersect either a specified query region or another topological feature. This can be performed with the help of the SDO_TOPO_ANYINTERACT operator.

```
SELECT * FROM city_streets
WHERE SDO_TOPO_ANYINTERACT(feature,
          query_sdo_geometry)="TRUE";
```

Note that there is no need for any explicit index creation in a topology. The RELATION$ table acts as the mapping table between features and topological primitive elements and is already indexed internally by the spatial engine. Spatial indexes are only necessary on the NODE$, EDGE$ and FACE$ tables and are maintained internally by Oracle Spatial.

Updating and Editing Topology As mentioned earlier, the main feature of creating and maintaining a topology is to ensure consistency and integrity for shared geometries. This means instead of updating the features directly, the users can update only the shared boundaries. Such update of shared elements will implicitly be reflected in all the features that share those updated elements. Typical functions provided for updates include addition/deletion of nodes, and addition/deletion of edges.

Key Applications

The SDO_GEOMETRY type is used to store geometry data in a number of applications [2]. Some examples include:

- Cadastral and census databases: all companies need to manage the location and shape of different assets (facilities, property). Likewise, government agencies such as Census Bureau maintain the exact shapes of administrative, geographical boundaries of different entities such as cities, counties and states. Other agencies manage the precise extent of individual properties and/or other land, natural resources. For each of these applications, there is an explicit need for storing large numbers of spatial geometries. The SDO_GEOMETRY type is ideal for such applications.
- Routing and navigation applications: a common use of spatial databases is route computation. The routing application server can store the streets, and highways inside the database and return the closest or fastest routes between two endpoints specified by a user.
- Mapping applications: several applications exist to construct maps from SDO_GEOMETRY data stored inside an Oracle database. Such maps can be integrated with other services for explorative analysis or for enabling business intelligence.

- Location-based services: Wireless data services increasingly use location data to enrich the user experience and provide valuable services. Uses include personal navigation systems, friend finders, panic button, roadside emergency, location-based yellow page searches, and the like. Locations of such wireless users can be stored and managed using the SDO_GEOMETRY type inside a database.
- Utility (electric, telephony) applications: the location and shape of electric and telephone networks can be effectively managed using the SDO_GEOMETRY type inside an Oracle database. Additional functionality in Oracle Spatial such as Linear Referencing can be utilized to effectively traverse such utility networks.
- Location-analysis applications: location analysis is frequently used in a variety of industries including banking, finance and insurance. For example in an insurance application, the risk factors for an entity (such as a person or a car) partially depend on its location. Management of the location of different entities using SDO_GEOMETRY and performing clustering analysis to identify high risk clusters, or to classify entities into different risk classes is a typical application in the insurance industry.

The SDO_TOPO_GEOMETRY is used in land management applications in local, state and federal governments. This model is useful for maintaining the integrity of land parcels and other territorial boundary management applications. The US Census Bureau and the Ordnance Survey model the land parcels using the SDO_TOPO_GEOMETRY type.

Future Directions

To summarize, Oracle Spatial offers two models for storing geometry data: the SDO_GEOMETRY and the SDO_TOPO_GEOMETRY models. The SDO_TOPO_GEOMETRY model is well suited for update-intensive applications. The SDO_GEOMETRY model may perform better if the majority of the workload consists of queries that fetch the entire geometry.

The types of geometries stored and processed in the SDO_GEOMETRY and the SDO_TOPO_GEOMETRY models are currently restricted to being 2D in nature. In future releases, Oracle Spatial may include support for storing 3D geometries (e. g., 3D solids) that appear in 3D city models and other advanced GIS applications. In addition to storing vector geometries, Oracle Spatial also supports efficient storage and retrieval of raster data. See Cross References for more details.

Cross References

► Raster Data

Recommended Reading

1. Kothuri, R., Ravada, S.: Efficient processing of large spatial queries. In: Proceedings of the Symposium of Spatial and Temporal Databases, Redondo Beach, CA, 12–15 July 2001
2. Kothuri, R., Godfrind, A., Beinat, E: Pro Oracle Spatial. Apress, Berkeley, CA (2004)
3. Kothuri, R., Ravada, S., Sharma, J., Banerjee, J.: Indexing medium dimensionality data in Oracle. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. Philadelphia, PA, 1–3 June 1999
4. Beckmann, N., Kriegel, H., Schneider, R., Seeger, B.: The R* tree: An Efficient and Robust Access method for points and rectangles. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. Atlantic City, NY 23–25 May, 1990
5. Samet H.: The Design and Analysis of Spatial Data Structures. Addison-Wesley, Reading (1989)
6. Brinkoff, T., Kriegel, H.-P., Schneider, R., Seegar, B.: Multi-step processing of spatial joins. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, Minneapolis, MN, 24–27 May 1994
7. An, N., Kothuri, R., Ravada, S.: Improving performance with bulk inserts in Oracle R-trees. In: Proceedings of the International Conference on Very Large Data Bases (VLDB), Berlin, Germany, 9–12 Sept 2003
8. Oracle. (2006) 10g Release 2 Spatial User's Guide and Reference. http://www.oracle.com/products/spatial. Accesed 2006
9. White, D.A., Jain, R.: Similary Indexing Algorithms and Performance. In: Proc. SPIE. 2670, pp. 62–73 (1996)

Oracle Spatial GeoRaster

► Oracle Spatial, Raster Data

Oracle Spatial, Raster Data

QINGYUN (JEFFREY) XIE
Server Technologies & Location-Based Spatial Products, Oracle USA, Nashua, NH, USA

Synonyms

Oracle Spatial GeoRaster; DEM; Image; Field data; SQL; Gridded data

Definition

Oracle Spatial supports raster data storage and management within the Oracle RDBMS (Relational Database Management System) server. This Oracle database feature is called GeoRaster, which lets you store, index, query, analyze, and deliver raster image and gridded data and its associated metadata. GeoRaster provides a native SQL (Structured Query Language) data type and an object-relational schema. More specifically, it provides the SDO_GEORASTER data type, which can be used to define one or more columns of any user table. Each image or a multilayer grid, such as multispectral or hyperspectral images, DEM (Digital Elevation Model) and thematic raster maps, is then stored as a GeoRaster object in a row of that column in that table. The GeoRaster objects can be referenced to positions on the Earth's surface or in a local coordinate system. The user-defined table containing one or more SDO_GEORASTER columns is called a GeoRaster table and a GeoRaster database is basically a list of such GeoRaster tables in one or more user schemas.

Historical Background

Numerous remote sensors of different types on various platforms are collecting real-time image data about the Earth and our environment for different purposes on a daily basis. Individual images can be very small, but they can be huge on the scale of terabytes as well. A raster database could have millions of such images or other types of raster gridded datasets and so the total size of such databases could be on the scale of petabytes and even exabytes. The major challenge is how to securely and reliably manage those data, tightly integrate them with other business data, and to make sure such a management system is truly scalable and highly performing.

The traditional raster data management systems are typically built on file systems directly. Some of them take a hybrid approach, which stores metadata and attributes in an RDBMS system but still stores the images as files in file systems. However, most file formats support only limited image size and have rigid structures, which prohibit flexible and scalable storage, fast querying and complex manipulations. On the other hand, the file system itself cannot offer such image datasets good enough security, reliability, availability and manageability.

Another approach is to take advantages of a standard RDBMS system's large object (LOB) data types by loading the raster files into database binary LOB (BLOB) objects so that both metadata and raster data are stored inside the database. Since the raster data is still stored in some specific file formats, the aforementioned disadvantages of the file formats are automatically inherited in such a database system. Mixing various file formats together and dealing with them inside the database system can only add one more layer of complexity. In addition, scalability, performance and manageability are still the bottlenecks.

Oracle GeoRaster, which was first released in Oracle database 10g, takes a completely different approach by taking full advantage of the Oracle enterprise RDBMS system as well as making the raster data storage independent of the numerous rigid raster file formats. Basically, the Oracle database system is internally enhanced to provide a new GeoRaster data type and a new mechanism to store and manage raster datasets with virtually no limitations [1]. The unique data type and architecture of GeoRaster provide not only high standard security, reliability and availability, but also great manageability, virtually unlimited scalability with regard to both data volume and the number of concurrent users, and high-speed data processing and distribution, to name just a few advantages. Besides leveraging the traditional benefits of RDBMS and providing full integration of spatial raster data with other enterprise data types, GeoRaster also takes advantages of advanced grid computing technologies, offering the benefits of lower cost, higher quality, and greater flexibility [2]. As a raster database management system, GeoRaster provides optimal image and raster data management solutions for the geospatial industry as well as enabling better business applications and massive spatial data usage.

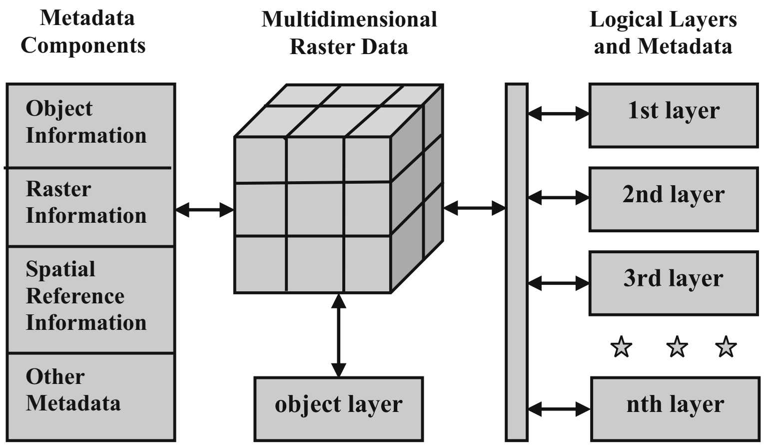
Scientific Fundamentals

In the geospatial area, raster data typically refers to two types of data: grid-based data and image data. Grid-based data or *gridded data* is a general term used for raster data. It's a rectangular grid of cells overlying an area. Each cell in the grid has the same size; this size is the resolution of the grid. Grid data stores either attribute values or attribute index values for each cell in the grid. The attribute index values are typically used to look up attribute values in a raster data table (RAT) or a value attribute table (VAT). Examples include digital terrain elevation, land use and land cover (LULC) information, pollution concentration, geological information, rainfall information, and many others. Digital imagery is a specialized type of raster data. It is a two-dimensional (2D) array (a matrix or grid) of regularly spaced picture elements (pixels). An image is created from optical or digital sensors, and is originally collected using a variety of technologies including satellite remote-sensing technologies and airborne photogrammetry. The size of the pixel is referred to as the resolution of the image. Digital images can be composed of many bands, referred to as multispectral or hyperspectral. GeoRaster supports both grid-based raster data and geographic images.

Logical Data Model

To best understand how Oracle GeoRaster stores and manages raster data, the first step is to understand its data model. It uses a component-based logically layered multidimensional raster data model [1,3]. Figure 1 shows the logical raster data model.

A raster data object consists of two parts: (1) raster cell data and (2) associated metadata. The raster cell data is



Oracle Spatial, Raster Data, Figure 1 The component-based logically layered multidimensional GeoRaster data model

a multidimensional matrix of raster cells. Each cell stores a value, referred to as the cell value. The number of bits used to store the cell value is called the cell depth. The matrix has a number of dimensions, a cell depth, and a size for each dimension. As a multidimensional matrix, the core data can be blocked for optimal storage, retrieval and processing.

In the GeoRaster data model, all associated information (other than the raster cell matrix) for the raster object is stored as "metadata". The GeoRaster metadata is divided into different components that contain the following information:

- Object information
- Raster information
- Spatial reference system information
- Date and time (temporal reference system) information
- Spectral (band reference system) information
- Layer information for each layer

In GeoRaster, each logical layer is a multidimensional matrix of cells. For example, for multichannel remote sensing imagery, the layers are used to model the bands of the imagery. Each band is one layer. The whole matrix of cells is called the object layer, which consists of one or more logical layers (or sublayers). By logically defining the layers, both gridded data and imagery can be unified in one data model and any metadata specifically associated with each individual layer can be well organized.

The object information includes metadata such as description and version information. The raster information includes metadata about the raster matrix, such as cell depth (e. g., 1BIT, 32BIT_S, or 64BIT_REAL), dimensionality and dimension sizes, blocking, interleaving, compression and information about pyramids. The spatial reference system metadata contains information required for georeferencing, in which a generic polynomial georeferencing model is defined. Value/raster attribute tables can be used to store multiple attribute information associated with the cell values of each layer (e. g., elevation values, and LULC types). In addition, scaling factors, colormap, grayscale information, statistics, histogram and other layer-based attributes could be stored as layer information metadata.

Physical Data Model

The GeoRaster data model is embodied as two native SQL data types and an object-relational schema inside Oracle RDBMS.

At the top level, one raster data (an image or a grid) is stored in Oracle as an object of the SDO_GEORASTER data type, which is defined as:

```
CREATE TYPE sdo_georaster AS OBJECT (
  rasterType NUMBER,
  spatialExtent SDO_GEOMETRY,
  rasterDataTable VARCHAR2(32),
  rasterID NUMBER,
  metadata XMLType);
```

Raster metadata is stored in the metadata attribute of the SDO_GEORASTER type. It is an XML document using the Oracle XMLType data type. The metadata is stored according to the GeoRaster metadata XML schema defined by GeoRaster. The spatial extent (footprint) of a GeoRaster object is part of the metadata, but it is stored separately as the spatialExtent attribute of the GeoRaster object. This approach allows GeoRaster to take advantage of the Oracle Spatial SDO_GEOMETRY type and related capabilities, such as using spatial R-tree [4] indexing on GeoRaster objects and building huge global imagery databases. Another attribute of the SDO_GEORASTER type is the rasterType, which contains dimensionality information.

The actual raster cell data of large GeoRaster objects can be blocked into small subsets for optimal storage, retrieval and processing. The raster blocks of a GeoRaster object are stored in another table named by the rasterDataTable and identified by the rasterID, both of which are attributes of the SDO_GEORASTER object. The table referred to by rasterDataTable is an object table of type SDO_RASTER, which is defined as follows:

```
CREATE TYPE sdo_raster AS OBJECT (
  rasterID NUMBER,
  pyramidLevel NUMBER,
  bandBlockNumber NUMBER,
  rowBlockNumber NUMBER,
  columnBlockNumber NUMBER,
  blockMBR SDO_GEOMETRY,
  rasterBlock BLOB);
```

This object table is called a raster data table, or simply RDT table. Each block is stored in the RDT table as a BLOB, and a geometry object (of type SDO_GEOMETRY) is used to define the precise extent of the block. Each row of the table stores only one block along with the blocking information related to that block. Each GeoRaster object can have their own RDT or share the same RDT with other GeoRaster objects. A B-tree [4] index is built on the RDT using the blocking information.

Pyramids are subobjects of a GeoRaster object that represent the raster image or raster data at differing sizes and degrees of resolution. The same blocking scheme is used to support pyramids and the pyramids are stored in the same RDT table.

It is important to note that the raster cell data inside the RDT table is maintained and managed internally by GeoRaster automatically once it is created. Users only need to understand and deal with the SDO_GEORASTER object type.

Using standard SQL language, a user can define any table and create one or more columns inside that table using the SDO_GEORASTER type. In the Oracle database, a GeoRaster data table is any user-defined table that has at least one data column of type SDO_GEORASTER. It could have any number of additional columns of any other SQL data types. SDO_GEORASTER objects include metadata and information about how to retrieve GeoRaster cell data that is stored in an RDT table of type SDO_RASTER. GeoRaster manages the relationship between a GeoRaster object and its raster data table automatically using a dictionary, called the GeoRaster sysdata table. From a user perspective, a GeoRaster database is basically a list of tables, in which each image or raster grid is stored as a GeoRaster object in one row as shown in Fig. 2. It can contain unlimited number of GeoRaster objects in one or more schemas and each object can be terabytes in size [2,5]. Users can build appropriate indexes on various columns of the GeoRaster tables e. g., a spatial R-tree index on the raster extent and B-tree indexes on other columns so that queries and other operations on the tables can be supported efficiently.

Creating a GeoRaster Database

To illustrate how to create a GeoRaster database, the following sample statements show the easy-to-use features. Assuming a user wants to build an image database for a city. The following statement creates a GeoRaster table, in which all or some of the images will be stored.

```
CREATE TABLE city_images
  (image_id NUMBER,
   image_description VARCHAR2(50),
   image SDO_GEORASTER);
```

Then the user creates the GeoRaster DML trigger on the image table, which is required in Oracle 10g to maintain the GeoRaster sysdata table. In Oracle 11g, users can skip this step because the trigger is automatically created and managed by Oracle database server whenever a GeoRaster table is created.

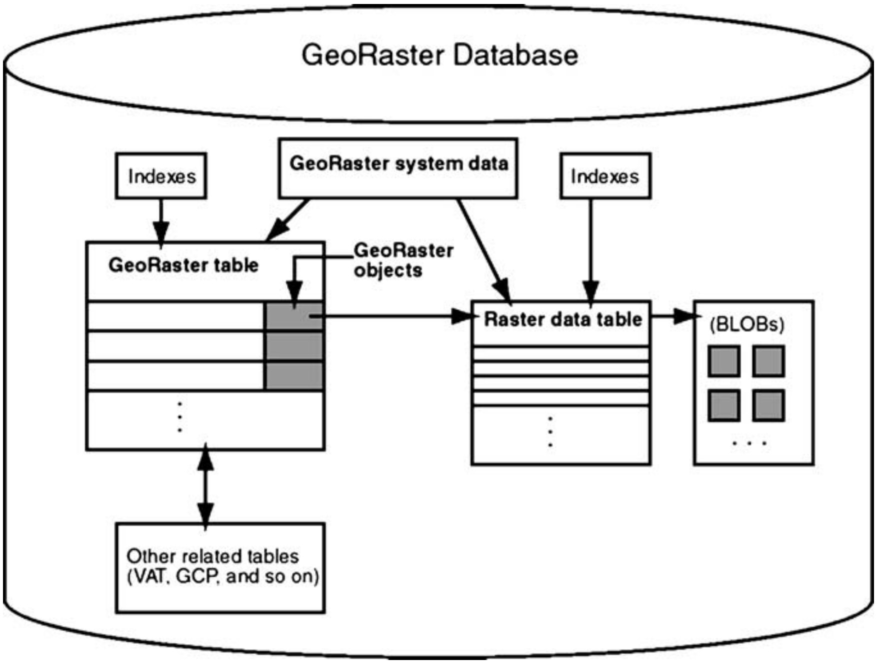
```
CALL sdo_geor_utl.createDMLTrigger(
  'city_images', 'image');
```

Users are required to create the RDT tables. For example,

```
CREATE TABLE city_images_rdt_1 OF SDO_RASTER
  (PRIMARY KEY (rasterID, pyramidLevel,
      bandBlockNumber, rowBlockNumber,
      columnBlockNumber))
   TABLESPACE tbs_1
   LOB(rasterBlock) STORE AS lobseg
     (TABLESPACE tbs_2
      CHUNK 32768
      CACHE READS
      PCTVERSION 0
      STORAGE (PCTINCREASE 0));
```


Instead of automatically creating the RDT tables, GeoRaster allows users to create them. By doing so, users gain full control of the RDT table creation so that appropriate tuning and partitioning can be applied to improve scalability and performance.

Now, users can upload all images into the city_images table. While all the images can be stored in only one RDT table, users can choose to create many RDT tables so that each RDT table would not grow too big. After initiating and inserting an empty GeoRaster object into the image table, users can use a lot of third-party tools (such as the PCI GeoRaster ETL tool and the Safe Software FME Oracle Edition) or use GeoRaster's client-side loader or the server-side importFrom procedure to load the images. For example, after granting read permission on the tm1.tif file to the database, users can call the importFrom procedure to load the image as follows:



Oracle Spatial, Raster Data, Figure 2 GeoRaster objects in an Oracle database

```
DECLARE
  geor SDO_GEORASTER;
BEGIN
  -- Initialize an empty GeoRaster object
  -- into which the external image is to be
  -- imported.
  INSERT INTO city_images VALUES
    (1, 'Landsat TM Image',
       sdo_geor.init('city_images_rdt_1', 1));
  -- Import the TIFF image.
  SELECT image INTO geor FROM city_images
    WHERE image_id = 1 FOR UPDATE;
  sdo_geor.importFrom(geor, NULL, 'TIFF',
    'file', '/data/tm1.tif');
  UPDATE city_images SET image = geor
    WHERE image_id = 1;
  -- commit the transaction and the loading
  -- is completed
  COMMIT;
END;
```

The images can be loaded individually or in batch. Once the images are loaded, users can georeference them, generate or assign spatial extents (footprints) and then optionally create spatial index on the image column. For example, the following statement creates a spatial index named city_images_idx on the spatial extents of the images using default values for all parameters:

```
CREATE INDEX city_images_idx
  ON city_images (image.spatialextent)
  INDEXTYPE IS MDSYS.SPATIAL_INDEX;
```

You can also create one or more other indexes, such as a function-based index on the GeoRaster metadata objects using the Oracle XMLType or Oracle Text document indexing functionality, a standard B-tree index on other columns of the GeoRaster table.

Users can create additional GeoRaster tables using the same approach. Each GeoRaster object (the images in this example) has a unique entry in the GeoRaster sysdata view across the whole database. This sysdata view could function as a special catalog on the whole image database as well.

Once the data are loaded into GeoRaster, users can manage it, tune it, perform spatial queries and various advanced processing as described in the next section.

Operations on GeoRaster Databases

Besides the standard enterprise database features, GeoRaster is tightly integrated with Oracle Spatial geometry features [4]. The spatial extent is of the SDO_GEOMETRY type, which could be in any projections or geodetic coordinate systems. The raster data could be located anywhere on the Earth and advanced spatial query and operators can help users quickly find any images of interest around the globe.

GeoRaster provides over 100 raster and metadata operations through a SQL API (Application Programming Language) to optimally manage GeoRaster data in support of various application requirements. Users can achieve a lot of goals using the following list of key operations:

- Adjust the internal raster blocking size to optimize the storage.

- Change interleaving types among BSQ, BIL and BIP.
- Change the cell depth, of which 1-bit to 32-bit integers and 32-bit and 64-bit floating numbers are supported.
- Generate or remove pyramids.
- Compress or decompress GeoRaster objects in lossless DEFLATE and lossy JPEG compression types. Other compressions are supported through third-party plug-ins. (10gR2)
- Edit and update raster data.
- Enlarge or shrink images.
- Mosaick large raster datasets.
- Merge GeoRaster objects or layers (11g).
- Crop images and perform subsetting to create new GeoRaster objects for persistent storage in the database or as an answer to a specific query for web distribution and display.
- Georeference a GeoRaster object using specified cell-to-model transformation coefficients.
- Generate a Spatial geometry that contains the spatial extent of the GeoRaster object.
- Add or remove a bitmap mask for a GeoRaster object or a GeoRaster layer (11g).
- Do statistical analysis and generate histograms (11g).
- Query, delete and update most items of the metadata through tens of functions.

The SQL API is provided for aiding in the development of and integration with applications on top of Oracle GeoRaster. Users can use Java, C or C++ to leverage this API or directly access the binary data of the open GeoRaster data model. GeoRaster has been extended, augmented or leveraged by partner technologies delivered as load/transform/export tools, comprehensive remote sensing and image processing client tools, or in the form of visualization engines.

Key Applications

GeoRaster can be used in broad application groups including geographic information system (GIS) and remote sensing applications, location-based business applications, and geoimage and gridded raster data repositories. The major domains of georaster include the following sciences and technologies. GeoRaster is used in numerous application areas that are related to or use these technologies.

Remote Sensing and Photogrammetry

The original data collected by remote sensing and photogrammetry is often called geoimagery, including satellite images and aerial photographs. The wavelength, number of bands, and many other factors determine the characteristics of the geoimages. The geoimages can have different cell depths and can be single-band, multiband, or hyperspectral. The main products from a photogrammetry system may include digital elevation models (DEMs) and orthoimagery. GeoRaster can be used to manage all these raster data sets, together with the georeferencing information. One of the biggest challenges in this area is the huge volume of such data. GeoRaster provides an effective, efficient and scalable approach to raster data archiving, management, processing and distribution [2].

Geographic Information Systems and Cartography

A GIS captures, stores, and processes geographically referenced information. GIS software has traditionally been either vector-based or raster-based. Raster-based GIS systems typically process gridded raster data. Gridded data can be discrete or continuous. Discrete data, such as political subdivisions and land use and land cover types, is usually stored as integer grids. Continuous data, such as elevation, aspect, pollution concentration, ambient noise level, and wind speed, is usually stored as floating-point grids. Each of such raster data can be stored as GeoRaster objects. The attributes of a grid layer are stored in a relational VAT or RAT table. The VAT or RAT can be stored in the Oracle database as plain tables. Their names can be registered within the corresponding GeoRaster object so that raster GIS applications can use the table. In addition, Oracle Spatial tightly integrates vector data with raster data, so providing a central storage and management solution in this area.

Cartography is the science of creating maps, which are two-dimensional representations of the three-dimensional Earth (or of a non-Earth space using a local coordinate system). Today, many maps are digitized or scanned into digital forms, which can be stored in raster format and thus can be managed by GeoRaster.

Geology, Geophysics, and Geochemistry

Geosciences, such as geology, geophysics and geochemistry, use digital data and produce many digital raster datasets that can be managed by GeoRaster. For example, in geology, the data includes regional geological maps, stratum maps, and rockslide pictures. In geological exploration and petroleum geology, computerized geostratum simulation, synthetic mineral prediction and 3-dimensional oil field characterization are widely used, all of which involve raster data. In geophysics, datasets about gravity and electromagnetic fields are saved and processed as rasters, along with georeferencing information. In geochemistry, the contents of multiple chemical elements can be analyzed and measured. The triangulated irregular network (TIN) technique is often used to produce raster maps for further analysis, and image processing is widely used.

Digital Image Processing

Digital image processing is used to process raster data in standard image formats, such as TIFF, GIF, JPEG, and Sun Raster, as well as in many geoimage formats, such as NITF, GeoTIFF, ERDAS IMG, and PCI PIX. Image processing techniques are widely used in remote sensing and photogrammetry applications. Most of the source, intermediate, and resultant imagery can be loaded and managed by GeoRaster.

Other Application Areas

Numerous application areas, which use the aforementioned technologies, could deploy GeoRaster with 3rd party analysis and visualization tools. These include state and local government, defense and security, emergency responses, enterprise asset management, agriculture monitoring, insurance risk assessment, to name only a few [6].

Future Directions

As a special type of database, Spatial database not only requires standard and advanced enterprise database features, but also needs to deal with complicated domain-specific data types, the metadata, spatial data analysis and distribution in support of various applications. With regard to raster spatial database technology, the trend is to continue to leverage enterprise RDBMS features and store and manage the raster data as native data types. Future directions for research and development include GeoRaster management utilizing advanced partitioning and parallelization features, sophisticated memory control techniques and multiuser concurrency enhancements, higher dimensional raster data support, standards-based metadata management, image content-based indexing and query, raster-based object recognition, raster map algebra and topological analysis, server-based geoprocessing and modeling, raster data streaming and caching.

Cross References

► Oracle Spatial, Geometries
► Raster Data

Recommended Reading

1. Murray, C., Blowney, J., Xie, J., Xu, T., Yuditskaya, S., Zhang, Z.: Oracle Spatial GeoRaster, 10*g* Release 2 (10.2). Oracle, Redwood Shores, CA, USA (2005)
2. Xie, J., Li, Z., Xu, W.: Using Enterprise Grid Computing Technologies to Manage Large-Scale Geoimage And Raster Databases. In: The Proceedings of ASPRS 2006 Annual Conference, Reno, Nevada, May 1–5, 2006
3. Xie, J., Sharma, J. (2005) Oracle Database 10*g*, Managing Spatial Raster Data Using GeoRaster—Technical White Paper. http://www.oracle.com/technology/products/spatial/index.html (2007). Accessed 1 February 2007
4. Kothuri, R., Godfrind, A., Beinat, E.: Pro Oracle Spatial. Apress, Berkley, CA, USA (2004)
5. Xie, J., Li, Z., Xu, T. (2005) Oracle Database 10*g* GeoRaster: Scalability and Performance Analysis. http://www.oracle.com/technology/products/spatial/index.html (2007). Accessed 1 February 2007
6. Xie, J., Sharma, J. (2005) Oracle Database 10*g*, Managing Spatial Raster Data Using GeoRaster—Business White Paper. http://www.oracle.com/technology/products/spatial/index.html (2007). Accessed 1 February 2007

Organization of IT Infrastructure Based On Services

► Service-Oriented Architecture

Organizing Concept

► Wayfinding, Landmarks

Ortho-Image

► Photogrammetric Products

Ortho-Mosaic

► Photogrammetric Products

Outlier

► Homeland Security and Spatial Data Mining

Outlier Detection

YUFENG KOU, CHANG-TIEN LU
Department of Computer Science, Virginia Tech,
Falls Church, VA, USA

Synonyms

Anomaly detection

Definition

Although there is no consensus on how to describe outliers, Barnet's definition is accepted by many statisticians and computer scientists, describing an outlier as "one observation that appears to deviate markedly from other members of the sample in which it occurs [1]." In recent years, outlier detection has begun to be widely used in numerous applications. In different applications, outliers may have different names such as anomalies, deviations, exceptions, faults, and irregularities. Outlier detection can help identify intrusions in computer networks, locate malfunctioning parts in a manufacture streamline, pinpoint suspicious usage of credit cards, and monitor unusual changes of stock prices.

Main Text

Rare events often reveal more important information than common ones, especially in case of the computer security, emergency responses, and medical diagnoses. For example, a web server is expected to handle thousands of connections from end-users, but in order to identify malicious intrusions, it is necessary to be able to identify anomalous connections, which are exceptional in IP address, port number, connection duration, and login frequency. Abnormal behaviors of these connections may indicate attacks from hackers. Therefore, abnormal pattern recognition, or outlier detection, has attracted a great deal of attention from researchers and engineers and has become an indispensable branch of data mining.

According to Breunig's classification [2], the existing traditional outlier detection algorithms can be categorized into four categories: clustering-based, distribution-based, depth-based, and distance-based. Outliers emerge as a byproduct of clustering, thus a few clustering algorithms can be used to detect outliers. These algorithms identify objects as outliers if they are far away from any cluster. Since these algorithms are aimed to extract clusters, their efficiency and effectiveness may not be optimized for outlier detection.

Distribution-based methods use a standard distribution such as Normal or Gaussian to measure the data set and the data points deviating from this distribution are defined as outliers. Depth-based methods organize the data in different layers of *k-d* convex hulls where data in the outer layers tend to be outliers. However, these methods are not widely used due to their high computation costs for multi-dimensional data. Distance-based methods may be the most widely used techniques which define an outlier as a data point having an exceptionally far distance to the other data points. Breunig et al. proposed a "local density" based method [2] where the local density of an object is defined based on its *k*-distance neighborhood. For a given point, the lower its local density is, the higher its probability of being an outlier.

The above methods for detecting outliers focus on the case of a single attribute. For detecting outliers with multiple attributes, classic outlier detection approaches are ineffective due to the "curse of high dimensionality," i. e., the sparsity of the data objects in high dimensional space. Two categories of research work have been conducted in order to address this issue. The first is to project high dimensional data to low dimensional data and the second is to re-design distance functions to accurately define the proximity relationship between data points.

Cross References

► Outlier Detection, Spatial

Recommended Reading

1. Barnett, V., Lewis, T.: Outliers in Statistical Data. John Wiley, New York (1994)
2. Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: Lof: Identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, p. 93–104, Dallas, Texas, United States, 14–19 May 2000

Outlier Detection, Spatial

YUFENG KOU, CHANG-TIEN LU
Department of Computer Science, Virginia Tech,
Falls Church, VA, USA

O

Synonyms

Spatial anomaly detection

Definition

Spatial outliers or abnormal spatial patterns are those spatial objects whose non-spatial attribute values are markedly different from those of their spatial neighbors. The identification of spatial outliers can be used to reveal hidden but valuable knowledge in many applications. For example, it can help locate extreme meteorological events such as tornadoes and hurricanes, identify aberrant genes or tumor cells, discover highway traffic congestion points, pinpoint military targets in satellite images, determine possible locations of oil reservoirs, and detect water pollution incidents.

Historical Background

Data mining is a process used to dig out useful "nuggets of information" from large amounts of data stored either in databases, data warehouses, or other information repositories [5]. These "nuggets" can be used to identify the patterns that occur frequently, illustrate the interesting associations among different patterns, and classify data items based on their characteristics. Besides extracting general rules and frequent patterns, an equally important task of data mining is to identify abnormal patterns (outliers), which were often ignored or discarded as noises.

In recent years, the wide application of geographical information systems has generated huge amounts of spatial data. Due to the ever-increasing spatial data, knowledge discovery in spatial data has become an important research area [4,10]. Over the last twenty years, extensive research in this area has formed a new branch of data mining, known as spatial data mining. As defined in [6], spatial data mining is the process of discovering hidden but valuable patterns from large spatial data sets. Spatial data mining techniques can be classified into four categories: classification, clustering, association analysis, and outlier detection.

Spatial outlier detection aims to discover local instabilities which break the spatial autocorrelation and continuity. A spatial outlier is a spatially referenced object whose non-spatial attribute values are significantly different from those of other spatially referenced objects in its spatial neighborhood [10]. In contrast to traditional outliers, spatial outliers are local anomalies that are extreme compared to their neighbors [3], but do not necessarily deviate from the remainder of the whole data set. Informally, Spatial outliers can be called "local outliers" because they pay more attention to local differences, while traditional outliers can be called "global outliers" since they focus on global comparisons.

Although numerous traditional outlier detection methods have been developed, they cannot be directly applied to spatial data to extract abnormal patterns. This is because classical outlier detection is designed to process numerical and categorical data, whereas spatial data have more complex structures that contain extended objects such as points, lines, and polygons. Also, traditional outlier detection does not handle spatial relationships among input variables, while spatial patterns often exhibit continuity and high autocorrelations with nearby samples. As stated by the geographical rule of thumb, "Everything is related to everything else, but nearby things are more related than distant things [12]."

Scientific Fundamentals

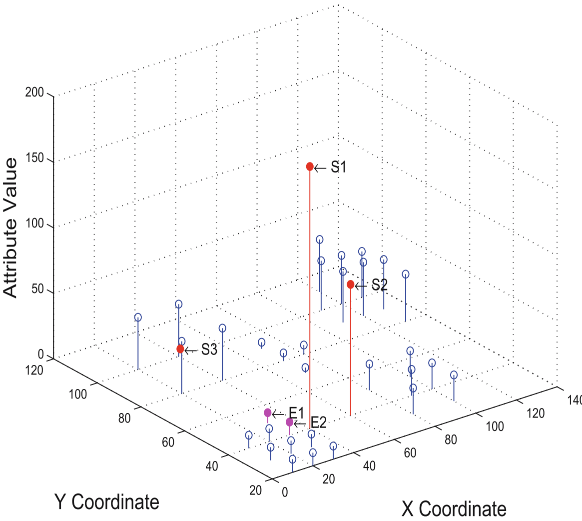
What is Spatial Data?

Spatial data refer to data objects pertaining to geometry and topology [9]. Geometry describes the shape, location, and size of an spatial object, while topology studies the spatial relationships among several objects, such as overlapping, adjacency, and inclusion. In contrast to non-spatial data, which simply deal with numbers and characters, spatial data are more complex and require special operations. First, spatial data are composed of complex structures such as points, lines, regions, and even 3-D objects. Moreover, spatial objects can be treated together to form even more complex objects. Second, due to their complex structures, the size of spatial data sets is much larger than that of non-spatial data sets. For example, thousands of (latitude, longitude) pairs may be required to represent the boundaries of a county. Third, spatial dependence, or spatial correlation, is common in many geospatial applications where the characteristics between neighboring objects tend to be similar. For instance, adjacent areas are likely to have similar temperature and humidity. Fourth, spatial data call for specific mechanisms for storage, indexing, and querying. Traditional database management techniques cannot be directly used for spatial data, since they do not support topological and geometric operations.

Spatial data can be categorized into two groups, macro-spatial data and micro-spatial data. Macro-spatial data are also referred to as geospatial data, as they contain spatial information for geological features. Geospatial data often exist in Geographic Information Systems (GIS), related to the earth environment or man-made architectures such as meteorological images, terrain maps, and mansion design figures. Micro-spatial data refer to those smaller objects whose characteristics are closely tied to their locations and shapes, for example, genetic sequences, Very Large-Scale Integration (VLSI) designs, and medical scan images. In many applications, spatial data not only relate to geometry and topology, but also contain temporal information. These data are called spatio-temporal data or spatial data sequences. Spatio-temporal data can be meteorological images continuously sent from satellites, real-time water quality data collected by monitoring stations, or traffic flow changes gathered by road detectors.

Spatial Outlier Detection

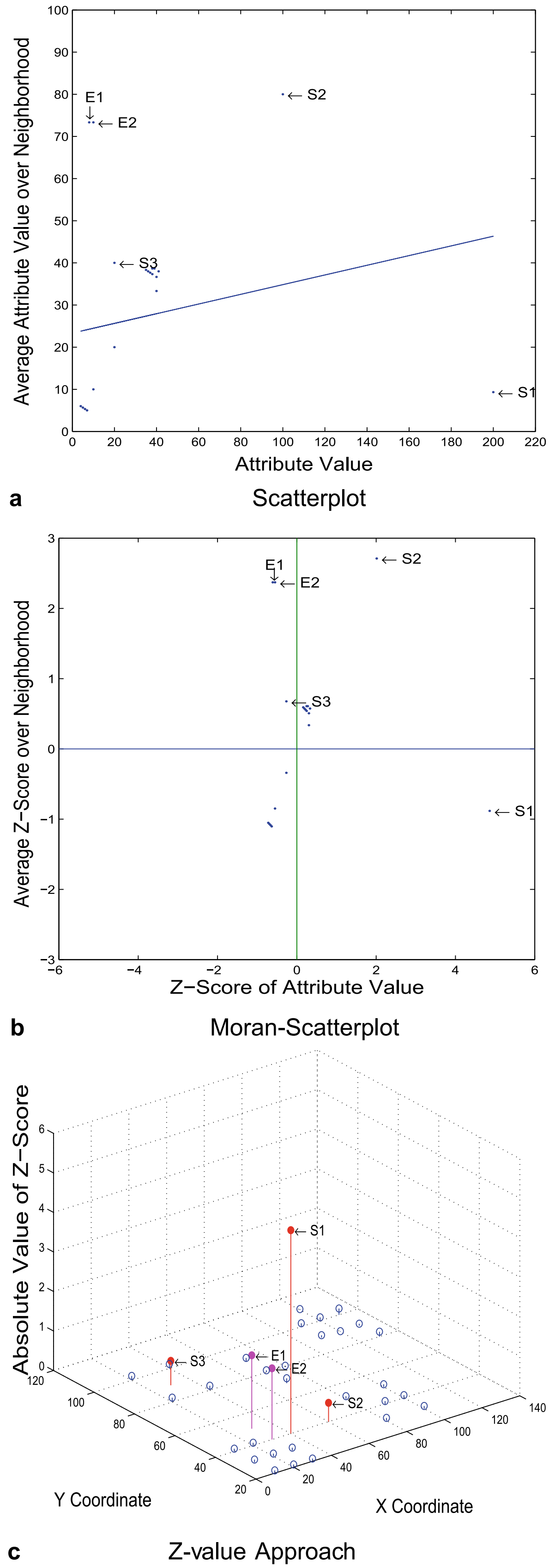
Spatial outlier detection is the process of identifying those spatial objects that are significantly different from their spatial neighbors. We use a simple example to illustrate what spatial outliers are. In Fig. 1, each point represents a spatial object. Objects are located in the $X-Y$ plane. The height of each vertical line segment represents the non-spatial attribute value of the corresponding object. If we define the number of spatial neighbors $k = 3$, point S1, S2, and S3 should be identified as spatial outliers since their non-spatial attribute values are much higher or lower than their spatial neighbors. For spatial outlier detection, spatial and non-spatial dimensions should be considered separately. The spatial dimension is used to define the



Outlier Detection, Spatial, Figure 1 A spatial data set

neighborhood relationship, while the non-spatial dimension is used to define the similarity function for comparison. Based on the number of non-spatial attributes, existing detection methods can be classified to single (non-spatial) attribute spatial outlier detection and multiple (non-spatial) attribute outlier detection.

The most widely used single attribute outlier detection methods include Scatterplot, Moran Scatterplot, and Z-value based method. Scatterplot and Moran Scatterplot are based on graphic visualization. In a Scatterplot [1], the X-axis represents the attribute values and the Y-axis denotes the average of the attribute values in the neighborhood. A least square regression line is used to identify spatial outliers. A scatter sloping upward to the right indicates a positive spatial autocorrelation; a scatter sloping upward to the left indicates a negative spatial autocorrelation. Nodes far away from the regression line are flagged as possible spatial outliers since they break the spatial autocorrelation and continuity. A Moran scatterplot [2] plots the normalized attribute value against the neighborhood average of normalized attribute values. A Moran scatterplot contains four quadrants. The objects located in the upper left and lower right quadrants are identified as spatial outliers. The upper left quadrant indicates low values surrounded by high value neighbors and the lower right quadrant suggests high values surrounded by low value neighbors.



Outlier Detection, Spatial, Figure 2 The Scatterplot, Moran-scatterplot, and Z-value of the data set shown in Fig. 1

Z-value approach uses the normalized difference between an object and its spatial neighbors as the measure of outlierness. If the difference is larger than a given threshold θ, the object will be marked as a spatial outlier. The Z-value is calculated as $Z_{s(x)} = |\frac{S(x)-\mu_s}{\sigma_s}|$, where $S(x)$ is the difference between the non-spatial attribute value of the object x and the average non-spatial attribute value of x's neighbors, μ_s is the mean value of $S(x)$, and σ_s is the standard deviation of $S(x)$ for the whole data set. Z-value approach provides satisfactory performance if $S(x)$ follows normal distribution. As proven in [11], $S(x)$ values are normally distributed if the non-spatial attribute values are normally distributed over the data set. The selection of the threshold θ is determined by the confidence level of $S(x)$.

One drawback of the above algorithms is that regular spatial points could be falsely detected as spatial outliers due to the presence of neighboring points with very high/low attribute values. A consequence of this disadvantage is that true spatial outliers could be ignored due to falsely detected spatial outliers if the expected number of spatial outliers is limited. These problems can be illustrated using the example data set shown in Fig. 1. Assuming the expected number of spatial outliers is 3 and the number of spatial neighbors k is taken to be 3, then we can easily observe that points $S1$, $S2$ and $S3$ are spatial outliers, since their attribute values are significantly different from those of their neighbors. However, the obtained result from running the Z-Value algorithm indicates that $S1$, $E1$ and $E2$ are spatial outliers, as shown in Fig. 2c. This detection error is mainly due to the large attribute value difference between point $S1$ and its neighboring points. For example, since $S1$ is inside the neighborhood of $E1$, the neighborhood function at $E1$ obtains a value much larger than the attribute value of $E1$, so that $E1$ is taken as an outlier. In general, the Z-value algorithm may potentially lead to some true spatial outliers being ignored and some false spatial outliers being identified. This disadvantage is also shared by other existing detection approaches such as Scatterplot 2a and Moran scatterplot 2b.

To amend the above issues, two iterative algorithms and one median-based algorithm are proposed [7]. The iterative algorithms, iterative-Z and iterative-R, detect spatial outliers through multiple iterations and in each iteration only one outlier is identified. The attribute value of this outlier is then replaced with the average of its neighbors, so that this outlier will not significantly impact the subsequent iterations. The difference between the iterative-Z and iterative-R is that the former selects the objects with the largest Z-value as outliers and the latter chooses the objects with the largest ratio between its non-spatial attribute and the average of its neighbors. The median-based method is a non-iterative algorithm. Different from the Z-value method, it uses the median instead of the mean to represent the average of a set of neighbors. The use of median potentially reduces the impact caused by the extreme neighboring points because median is a robust estimator of the center of a sample.

Multiple Attribute Spatial Outlier Detection

In many applications, there may be more than one non-spatial attributes associated with each spatial location. For example, in the Census data, each census tract contains several non-spatial attributes, including population, population density, income, poverty, housing, education, and race. Detecting outliers from these spatial data with multiple attributes will help demographers and social workers to identify local anomalies for further analysis. A popular method of multi-attribute spatial outlier detection is based on Mahalanobis distance.

Suppose $q(\geq 1)$ measurements (attribute values) are made on the spatial object x. We use a to denote the vector of these q values at x. Given a set of spatial points $X = \{x_1, x_2, \ldots, x_n\}$ in a space with dimension $p \geq 1$, an attribute function f is defined as a map from X to R^q (the q dimensional Euclidean space) such that for each spatial point x, $f(x)$ equals the attribute vector a. For convenience, we write $a_i = f(x_i) = (f_1(x_i), f_2(x_i), \ldots, f_q(x_i))^{\mathrm{T}}$ for $i = 1, 2, \ldots, n$. Denote the set $\{a_1, a_2, \ldots, a_n\}$ by A.

Let $NN_k(x_i)$ denote the k nearest neighbors of point x_i with $k = k(x_i)$ for $i = 1, 2, \ldots, n$. A neighborhood function g is defined as a map from X to R^q such that the jth component of $g(x)$, denoted $g_j(x)$, returns a summary statistic of the jth attribute values from all the spatial points inside $NN_k(x)$. For the purpose of detecting spatial outliers, all of the components of a at x can be compared with the corresponding quantities from the neighbors of x. A comparison function h is a function of f and g, whose domain is X and range is in R^r with $r < qq$. Examples of h include $h = f - g$, a map from X to R^q with $r = q$, and $h = f_1/g_1$, a map from X to R with $r = 1$. Denote $h(x_i)$ by h_i. A point x_i is an S-outlier if h_i is an extreme point of the set $\{h_1, h_2, \ldots, h_n\}$.

Under certain conditions, $h(x)$ follows a multivariate normal distribution. And if $h(x)$ is distributed as $N_q(\mu, \Sigma)$, i. e., q-dimensional vector $h(x)$ follows a multivariate normal distribution with mean vector μ and variance-covariance matrix Σ, then $(h(x) - \mu)^{\mathrm{T}} \Sigma^{-1} (h(x) - \mu)$ is distributed as χ_q^2, where χ_q^2 is the chi-square distribution with q degrees of freedom. Therefore, the probability that $h(x)$ satisfies $(h(x) - \mu)^{\mathrm{T}} \Sigma^{-1} (h(x) - \mu) > \chi_q^2(\alpha)$ is α, where $\chi_q^2(\alpha)$ is the upper (100α)th percentile of a chi-square distribution with q degrees of freedom. Thus intuitively, for a point x, if $(h(x) - \mu)^{\mathrm{T}} \Sigma^{-1} (h(x) - \mu)$ is sufficiently large, x should be treated as an S-outlier candidate.

Now suppose there are n spatial referenced objects $x_1, \ldots, x_n$. For the sample $h(x_1), \ldots, h(x_n)$, calculate the sample mean

$$\mu_s = \frac{1}{n}\sum_{i=1}^{n} h(x_i)$$

and sample variance-covariance matrix

$$\Sigma_s = \frac{1}{n-1}\sum_{i=1}^{n}[h(x_i) - \mu_s][h(x_i) - \mu_s]^{\mathrm{T}}.$$

Then we should expect that the probability of $h(x)$ satisfying $(h(x) - \mu_s)^{\mathrm{T}}\Sigma_s^{-1}(h(x) - \mu_s) > \chi_q^2(\alpha)$ is roughly α. Set $d(x) = [(h(x)-\mu_s)^{\mathrm{T}}\Sigma_s^{-1}(h(x)-\mu_s)]^{1/2}$. Therefore, if $d(x)$ or $d^2(x)$ is unusually large, x will be teated as a spatial outlier. In practice, it is possible that the inverse Σ_s^{-1} does not exist. If this happens, the Moore-Penrose generalized inverse Σ_s^{+} can be used to replace Σ_s^{-1}.

The quantity $d(x)$ defined above is actually the Mahalanobis distance from $h(x)$ to μ_s. The Mahalanobis distance provides a suitable way to identify points which are far from all of the others in a multidimensional space. It has been widely used in discriminant analysis, clustering, and principal analysis. It has many advantages over Euclidian distance when dealing with multivariate data. For example, the basic Euclidian distance treats each variable as equally important in calculating the distance, while Mahalanobis distance automatically accounts for the scaling of the coordinate axes.

Region Outlier Detection

In certain situations, an outlier may not appear as a single spatial object but in the form of a group of adjacent objects, i. e., a region. The spatial objects within the region have similar characteristics, and their characteristics are significantly different from the objects outside the region. We define this type of outlier as a "region outlier." Region outliers can be tornadoes in meteorology, traffic congestions in highway systems, and crude oil reservoirs in geology.

Several approaches have been developed to identify region outliers. For example, Lu et al. (2004) apply wavelet transformation to meteorological data to bring up distinct patterns that might be hidden within the original data [8]. Then a powerful image processing technique, edge detection with competitive fuzzy classifier, is used to identify the boundary of a region outlier. After that, to determine the center of the region outlier, the fuzzy-weighted average of the longitudes and latitudes of the boundary locations is computed. By linking the centers of the outlier regions within consecutive frames, the movement of a region outlier can be captured and traced.

Key Applications

Meteorological Data Analysis

In the field of meteorological data processing, spatial outliers are frequently associated with natural disasters such as tornadoes and hurricanes. Usually, meteorological data can be represented by digital images where the row and column correspond to the latitude and longitude and the color intensity represents the value of weather parameters. Research study has been conducted to detect and track regional outliers in a sequence of meteorological data frames by applying image processing techniques [8].

Medical Image Analysis

X-ray projection, ultrasound scan, magnetic resonance imaging, and gamma camera generate huge amounts of medical images. Spatial outlier detection methods can help medical personnel locate the abnormal regions in the images such as tumors and infected tissues.

Census Investigation

Census investigation is an important geospatial application. Spatial outlier detection can help identify abnormal patterns in census tracts, counties, and metropolitan areas. For example, we can determine the unusual correlation between levels of air pollution with housing prices, pinpoint crime hot-spots in big cities, or discover infrequent pupil-teacher ratios.

Contagious Disease Monitoring

Spatial outlier detection can assist public health staff in discovering the regions with abnormal disease reports. For example, if the number of West Nile virus cases in a county deviates much from those of its neighboring counties, this county should be investigated for conditions that facilitate or restrain the spread of the virus.

Transportation System

Transportation systems collect huge amount of traffic data including volume, speed, and occupancy from numerous monitoring stations along highways and local streets. Spatial outlier detection can help identify malfunctioning road detectors and locate traffic incidents and congestions.

Future Directions

Although spatial outlier detection has attracted tremendous attention from computer scientists, many challenges still need to be further studied, such as 3D outlier detection and outlier detection in spatio-temporal data streams. In

3D context, it is difficult to accurately define neighborhood relationship and delineate the boundary of a region. Several issues need to be investigated, such as 3D image processing and its combination with data mining techniques. The progress in this direction will assist medical personnel in extracting abnormal patterns from the large amount of ultrasonic scan and magnetic resonance images.

Many spatial data are continuously collected and it is impossible to save all the data elements or scan them multiple times to detect outliers. In addition, multiple data streams may exist simultaneously and correlate to one another. Meanwhile, the concept of the geospatial stream may drift over time, say, both the attribute values and object locations may keep changing. Therefore, it is a challenging task to dynamically define neighborhood relationship and design efficient algorithms which can incrementally identify anomalies from multiple correlated geospatial data streams.

Cross References

► Outlier Detection

Recommended Reading

1. Anselin, L.: Exploratory spatial data analysis and geographic information systems. In: New Tools for Spatial Analysis, p. 45–54. Eurostat, Luxemburg (1994)
2. Anselin, L.: Local indicators of spatial association: Lisa. Geogr. Anal. **27**(2):93–115 (1995)
3. Cerioli, A., Riani, M.: The ordering of spatial data and the detection of multiple outliers. J. Comput. Graphical Stat. **8**(2):239–258 June 1999
4. Haining, R.: Spatial Data Analysis in the Social and Environmental Sciences. Cambridge University Press, Cambridge, UK (1993)
5. Han, J., Kamber, M.: Data Mining Concepts and Techniques. Morgan Kaufmann Publishers, CA, USA (2001)
6. Koperski, K., Adhikary, J., Han, J.: Spatial data mining: Progress and challenges. In: Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'96), p. 1–10, Montreal, Canada June 1996
7. Lu, C.-T., Chen, D., Kou, Y.: Algorithms for spatial outlier detection. In: Proceedings of the Third IEEE International Conference on Data Mining, p. 597–600, Melbourne, Florida, United States, 19–22 Nov 2003
8. Lu, C.-T., Liang, L.R.: Wavelet fuzzy classification for detecting and tracking region outliers in meteorological data. In: GIS '04: Proceedings of the 12th annual ACM international workshop on Geographic information systems, p. 258–265, Washington DC, USA, 12–13 Nov 2004
9. Rigaux, P., Scholl, M., Voisard, A.: Spatial Databases: With Application to GIS. 2nd edn. Morgan Kaufmann Publishers, San Francisco, CA, United States, May 2001
10. Shekhar, S., Chawla, S.: A Tour of Spatial Databases. Prentice Hall, Upper Saddle River, New Jersey (2002)
11. Shekhar, S., Lu, C.-T., Zhang, P.: A unified approach to detecting spatial outliers. GeoInformatica **7**(2):139–166 (2003)
12. Tobler, W.: Cellular geography. In: Philosophy in Geography, pages 379–386, Dordrecht Reidel Publishing Company, Dordrecht, Holland (1979)

OWL

► Geospatial Semantic Integration

OWS

► OGC Web Services

PAI

▶ Positional Accuracy Improvement (PAI)

Pandemics, Detection and Management

Christopher L. Barrett, Stephen Eubank,
Bryan Lewis, Madhav V. Marathe
Virginia Bioinformatics Institute, Virginia Tech,
Blacksburg, VA, USA

Synonyms

Epidemiology, computational; Epidemics; Public health; Information management systems; Distributed information systems

Definition

Epidemiology is the study of patterns of health in a population and the factors that contribute to these patterns. It plays an essential role in public health through the elucidation of the processes that lead to ill health as well as the evaluation of strategies designed to promote good health. Epidemiologists are primarily concerned with public health data, which includes the design of studies, evaluation and interpretation of public health data, and the maintenance of data collection systems.

Computational Epidemiology is the development and use of computer models for the spatio-temporal diffusion of disease through populations. The models may range from descriptive, e. g. static estimates of correlations within large databases, to generative, e. g. computing the spread of disease via person-to-person interactions through a large population. The disease may represent an actual infectious disease, or it may represent a more general reaction-diffusion process, such as the diffusion of innovation. The populations of interest depend on the disease, including humans, animals, plants, and computers. Similarly, the interactions that must be represented depend on the disease and the populations, including physical proximity for aerosol-borne disease, sexual contact for sexually transmitted diseases, and insect feeding patterns for mosquito-borne diseases. In general, then, computational epidemiology creates computer models of diffusive processes spreading across interaction networks.

The basic goal of epidemiological modeling is to understand the dynamics of disease spread well enough to control it. Potential interventions for controlling infectious disease include pharmaceuticals for treatment or prophylaxis, social interventions designed to change transmission rates between individuals, physical barriers to transmission, and eradication of vectors. Efficient use of these interventions requires targeting subpopulations that are on the critical path of disease spread. Computational models can be used to identify those critical subpopulations and to assess the feasibility and effectiveness of proposed interventions.

Historical Background

Epidemiology did not emerge as a distinct discipline until the mid-19th century as the medical sciences sought to determine the efficacy of different medical practices. John Snow famously interrupted the 1854 cholera outbreak in London by removing the handle of the Broad Street pump, an event that is widely credited with bringing epidemiology into the mainstream. His studies along with those of many others were responsible for bringing about wide-ranging public health reforms and laid the foundation for the development of the germ theory of disease causation. Once etiological agents were identified as the cause of disease, the sanitary reforms of the late 19th and early 20th centuries greatly reduced the incidence of infectious disease in the human population. Epidemiology continued to identify novel causes of disease but also matured and began to consider the social determinants of health. Aided by improved statistical tools epidemiologists were able to focus on more nuanced analysis of population-wide health data, for example, linking lung cancer to smoking. The pharmaceutical revolution of the mid-20th century required epidemiology to assess the efficacies of the new therapies being created. Epidemiology has further gained from the ability to manipulate large bodies of data and perform complex calculations on this data using computers. As technology has improved the sophistication of techniques to analyze public health data used by epidemiologists has kept pace. Recent years have seen the emergence of computational epidemiology which focuses on using complex computer models and innovative forms of data analysis. At its core, epidemiology is still focused on improving the health of the public, however, recently, epidemiologically inspired methods and techniques have been harnessed to analyze computer security, network routing, distributed databases, marketing strategies, and other social phenomena.

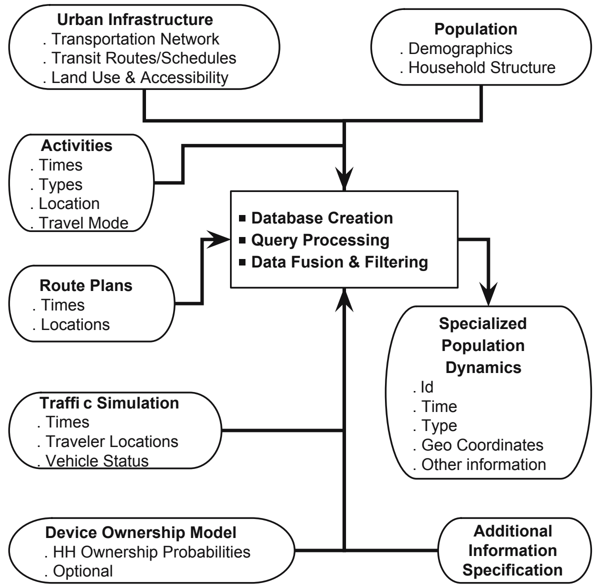
Scientific Fundamentals

The spread of infectious disease depends both on properties of the pathogen and the host. An important factor that greatly influences an outbreak of an infectious disease is the structure of the interaction network across which it spreads. Descriptive models are useful for estimating properties of the disease, but the structure of the interaction network changes with time and is often affected by the presence of disease and public health interventions. Thus generative models are most often used to study the effects of public health policies on the control of disease.

Aggregate or collective computational epidemiology models often assume a population is partitioned into a few subpopulations (e. g. by age) with a regular interaction structure within and between subpopulations. The resulting model can typically be expressed as a set of coupled ordinary differential equations. Such models focus on estimating the number of infected individuals as a function of time, and have been useful in understanding population-wide interventions. For example, they can be used to determine the level of immunization required to create herd immunity.

Disaggregate or individual-based models, in contrast, represent each interaction between individuals, and can thus be used to study critical pathways. Disaggregate models require neither partitions of the population nor assumptions about large scale regularity of interactions; instead, they require detailed estimates of transmissibility between individuals. The resulting model is typically a stochastic finite discrete dynamical system. For more than a few individuals, the state space of possible configurations of the dynamical system is so large that they are best studied using computer simulation.

GIS tools and techniques play an important role in building these computational tools. The overall approach followed by disaggregate models consists of the following steps:



Pandemics, Detection and Management, Figure 1 Schematic diagram showing how various databases and integrated to create a synthetic population. GIS plays an important role in constructing these synthetic populations

Step 1 Creating a set of (agent) synthetic interactors,
Step 2 Generating (time varying) interaction networks,
Step 3 Detailed simulation of the epidemic process.

Step 1 creates a synthetic urban populations [3,4,5,6], and is done by integrating a variety of databases from commercial and public sources into a common architecture for data exchange that preserves the confidentiality of the original data sets, yet produces realistic attributes and demographics for the synthetic individuals. Figure 1 shows a schematic diagram. The synthetic population is a set of synthetic people, each associated with demographic variables drawn from any of the demographics available in the census [3,7]. Joint demographic distributions can be reconstructed from the marginal distributions available in typical census data using an iterative proportional fitting (IPF) technique. Each synthetic individual is placed in a household with other synthetic people and each household is located geographically in such a way that a census of our synthetic population yields results that are statistically indistinguishable from the original census data, if they are both aggregated to the block group level. Synthetic populations are thus statistically indistinguishable from the census data; nevertheless, since they are synthetic they respect privacy of individuals within the population. Note that, census tables are precisely constructed so as to respect privacy. The *synthetic individuals* carry with them a complete range of demographic attributes collected in the census data. This include variables such as income level, age, etc.

In Step 2, a set of activity templates for households are determined based on several thousand responses to an activity or time-use survey. These activity templates include the sort of activities each household member performs and the time of day they are performed. Each synthetic household is matched with one of the survey households, using a decision tree based on demographics such as the number of workers in the household, number of children of various ages, etc. The synthetic household is assigned the activity template of its matching survey household. For each household and each activity performed by this household, a preliminary assignment of a location is made based on observed land-use patterns, tax data, etc. This guess must be calibrated against observed travel-time distributions. However, the travel-times corresponding to any particular assignment of activities to locations cannot be determined analytically. Using sophisticated techniques in combinatorial optimization, machine learning and agent based modeling the populations, their activity locations and their itineraries [3,5] are refined so as to be structurally and statistically consistent. See Fig. 1 for a schematic diagram. Thus for a city – demographic information for each person and location, and a minute-by-minute schedule of each person's activities and the locations where these activities take place is generated by a combination of simulation and data fusion techniques. This forms the basis of the interaction network that can be abstractly represented by a (vertex and edge) labeled bipartite graph G_{PL}, where P is the set of people and L is the set of locations. If a person $p \in P$ visits a location $l \in L$, there is an edge $(p, l, \text{label}) \in E(G_{PL})$ between them, where label is a record of the type of activity of the visit and its start and end points. Each vertex (person and location) can also have labels. The person labels correspond to his/her demographic attributes such as age, income, etc. The labels attached to locations specify the location's attributes such as its x and y coordinates, the type of activity performed, maximum capacity, etc. Note that, there can be multiple edges between a person and a location recording different visits.

Step 3 consists of developing computational model for representing the disease within individual interactor and its transmission between interactors. The model can be viewed as a *coupled probabilistic timed finite state machine*. Each individual is associated with a timed probabilistic finite state machine – the state transitions are probabilistic; the transitions may be timed – i. e. they may occur at a specified time after the previous transition – or there may be a fixed probability of transition for each discrete time interval. Furthermore, the automata are coupled to other automata – this coupling is derived from the social contact network. The state of the automata corresponding to an individual are updated probabilistically based on the current state of the individual and the disease state of his neighbors. This state transition is probabilistic and depends on the duration of contact. It may also depend on the attributes of the people involved (age, profession, health status, etc.) as well as the type of contact (intimate, casual, etc.), and it might not be symmetric (a child is more likely to infect a teacher than the other way around). Again GIS tools play an important role in constructing the models. The tools include: methods for integrating spatio-temporal surveillance data, mapping of the disease outbreak to geographic locations, designing intervention effective strategies, such as closing specific public locations, social distancing, etc.

An integrated information management system can now be constructed using these computational models. The systems is usually event triggered and the supporting system should ideally self organize in response to such a trigger. Figure 2 shows a possible architecture for such a system.

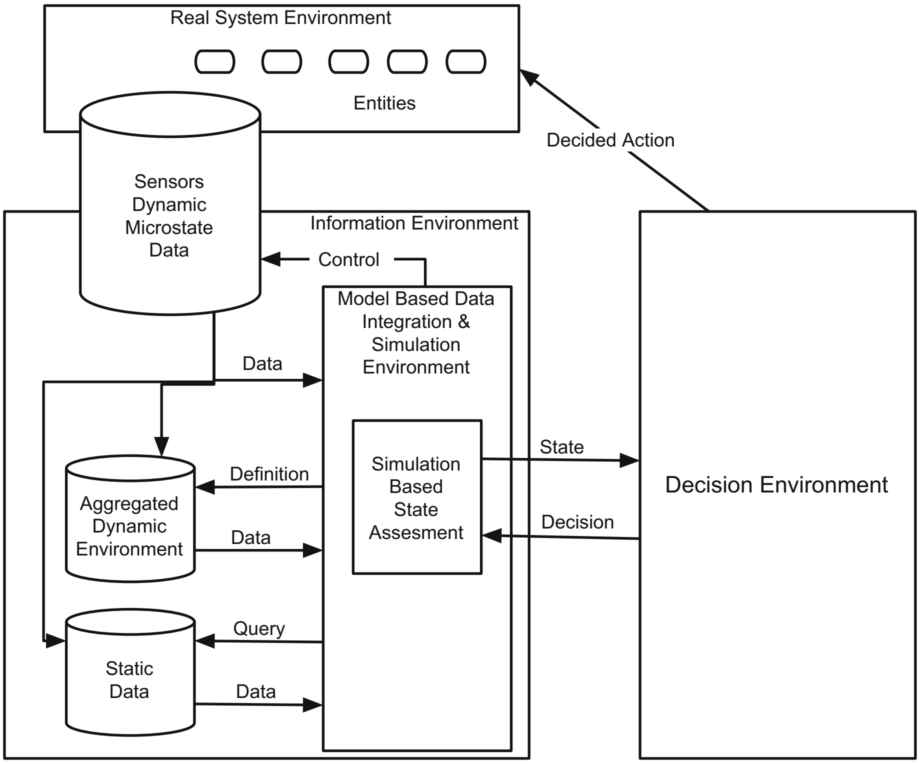
Application Areas

Epidemiological simulations are a subclass of more general interaction processes called reaction-diffusion processes. In their general form, such systems consist of a set of entities (interactors) and an interaction-hypergraph denoting neighborhood relationships among interactors. At each time step, based on certain criteria, subsets of neighboring interactors interact. The interactions can result in two things: (i) the state or the function of the interactors can change, the interaction-hypergraph can change. A number of applications in physical and social sciences that can be viewed in this framework. These include: (i) physical systems such as n-body dynamics, (ii) biological systems such as bio-chemical reactions, (iii) social systems such as diffusion of norms and fads, (iv) public health systems such as epidemics, (v) communication networks such as spread of worms on the Internet, routing of packets and updating distributed databases, (vi) business and information systems such as viral marketing, etc.

The systems differ in the relative rates at which interactions happen as compared to the change in the state of the interactors, and the network structure. For example, in epidemics, an individual when exposed to an infectious disease can become infected after a certain time period that depends on the demographic properties of the individual and the disease characteristics. When routing packets over a wireless ad-hoc network, the state of the interactor and the the connectivity of the underlying network changes rapidly.

Public Health

Computational epidemiology is a new and fast growing branch of public health. Using sophisticated and highly



Pandemics, Detection and Management, Figure 2 Schematic diagram for constructing a simulation based integrated information management system

tuned computer simulations, different public health interventions can be evaluated that would be unfeasible and/or unethical in the real world. Furthermore, as these techniques become more and more sophisticated, these population and disease dynamics can be better analyzed *in silico* than *in vivo*. See [6,8,12,13,14,15,17,18,23].

Social Sciences

Social Sciences have a rich history of studying social phenomenon using epidemic style models. This includes, diffusion of norms, fads, etc [21,26,29].

Epidemic/Viral Marketing and Advertising

This class of applications consist of marketing techniques to spread brand awareness. The information about brands, products, etc. can be exchanged via word of mouth using social networks that capture individuals meeting each other in physical or cyberspace (via e. g. blogs, chat rooms). Viral marketing is popular because it is usually easy and affordable to execute the marketing campaign. It is becoming all the more popular due to the Internet which allows for much more rapid dissemination.

Computer Network Security

Epidemic style models are being used to study the spread of worms and viruses on the Internet [2,27]. Computer models can be used to study the propagation of viruses as well as ways to control its spread. A unique feature of these systems is that unlike biological systems, computer viruses usually spread extremely fast, usually in a matter of hours if not minutes. Moreover, the viruses are synthetic; humans construct these viruses and their genetic variations.

Distributed Computing, Communication and Information Systems

A number of tasks in computing, communication and information systems can be achieved by using epidemic style algorithms. This includes: routing using local information under unreliable conditions [10], location of resources [11], and updating replicated distributed databases [9,11].

Acknowledgement

The work was supported in part by NIH MIDAS project.

Cross References

- ► Bioinformatics, Spatial Aspects
- ► Biomedical Data Mining, Spatial
- ► Data Analysis, Spatial
- ► Exploratory Spatial Analysis in Disease Ecology

Recommended Reading

1. Albert, R., Barabasi, A.: Statistical mechanics of complex networks. Rev. Mod. Phys. **74**, 47–97 (2002)
2. Boguna, M., Pastor-Satorras, R., Vespignani, A.: Epidemic Spreading in Complex Networks with Degree Correlations. In: Pastor-Satorras, R., Rubi, M., Diaz-Guilera, A. (eds.) Statistical Mechanics of Complex Networks. Lect. Notes Phys. **625**, 127–147 (2003)
3. Barrett, C., Beckman, R., Berkbigler, K., Bisset, K., Bush, B., Campbell, K., Eubank, S., Henson, K., Hurford, J., Kubicek, D., Marathe, M., Romero, P., Smith, J., Smith, L., Speckman, P., Stretz, P., Thayer, G., Eeckhout, E., Williams, M.D.: TRANSIMS: Transportation Analysis Simulation System. Technical Report LA-UR-00–1725, Los Alamos National Laboratory Unclassified Report. An earlier version appears as a 7 part technical report series LA-UR-99-1658 and LA-UR-99-2574 to LA-UR-99–2580 (2001)
4. Beckman, R.J., et al: TRANSIMS-Release 1.0: The Dallas-Fort Worth Case Study, Technical Report LA-UR-97-4502, Los Alamos National Laboratory (1997)
5. Barrett, C., Eubank, S., Marathe, M.: Modeling and Simulation of Large Biological, Information and Socio-Technical Systems: An Interaction Based Approach, to appear. In: Goldin, D., Smolka, S., Wegner, P. (eds.) Interactive Computation: The New Paradigm. Springer, New York (2005)
6. Barrett, C., Smith, J.P., Eubank, S.: Modern Epidemiology Modeling. Sci. Am. **292**(3), 54–61 (2005)
7. Beckman, R., Baggerly, K., McKay, M.: Creating synthetic baseline populations. Transportation Research Part A, Policy Pract. **30**, 415–429 (1996)
8. Carley, K., Fridsma, D., Casman, E., Altman, N., Chang, J., J., Kaminski, B., Nave, D., Yahja, A.A.: BioWar: Scalable Multi-Agent Social and Epidemiological Simulation of Bioterrorism Events. NAACSOS conference proceedings, Pittsburgh, PA (2003)
9. Kempe, D., Kleinberg, J.M., Demers, A.J.: Spatial gossip and resource location protocols. J. ACM **51**(6), 943–967 (2004)
10. Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J., Silva, F.: Directed Diffusion for Wireless Sensor Networking. ACM/IEEE Trans. Netw. **11**(1), 2–16 (2002)
11. Demers, A.J., Greene, D.H., Hauser, C., Irish, W., Larson, J., Shenker, S., Sturgis, H.E., Swinehart, D.C., Terry, D.B.: Epidemic Algorithms for Replicated Database Maintenance. PODC 1–12 (1987)
12. Eubank, S., Guclu, H., Anil Kumar, V.S., Marathe, M., Srinivasan, A., Toroczkai, Z., Wang, N.: Modeling Disease Outbreaks in Realistic Urban Social Networks. Nature **429**, 180–184 (2004)
13. Eubank, S., Anil Kumar, V.S., Marathe, M., Srinivasan, A., Wang, N.: Structure of Social Contact Networks and their Impact on Epidemics. In: AMS-DIMACS Special Volume on Epidemiology **70**, 181–213 (2005)
14. Ferguson, N.L., Cummings, D.A.T., Cauchemez, S., Fraser, C., Riley, S., Meeyai, A., Lamsirithaworn, S., Burke, D.S.: Strategies for containing an emerging influenza pandemic in Southeast Asia, Nature **437**, 209–214 (2005)
15. Ferguson, N.L., Cummings, D.A.T., Fraser, C., Cajka, J.C., Cooley, P.C., Burke, D.S.: Strategies for mitigating an influenza pandemic, Nature, 448–452 (2006)
16. Ganesh, A., Massoulie, L., Towsley, D.: The Effect of Network Topology on the Spread of Epidemics. *IEEE Infocom* 2005, Miami, FL (2005)
17. Germann, T.C., Kadau, K., Longini Jr., I.M., Macken, C.A.: Mitigation strategies for pandemic influenza in the United States. Proc. of National Academy of Sciences (PNAS), vol. 103(15), 5935–5940 (2006)
18. Hethcote, H.: The Mathematics of Infectious Diseases. SIAM Rev. **42**(4), 599–653 (2000)
19. Jurvetson, S.: What exactly is viral marketing? Red Herring **78**, 110–112 (2000)
20. Kephart, J.O., White, S.R.: Directed-graph epidemiological models of computer viruses. Proc. 1991 IEEE Computer Society Symposium on Research in Security and Privacy, pp. 343–359 (1991)
21. Macy, M., Willer, R.: From Factors to Actors: Computational Sociology and Agent-Based Modeling. Ann. Rev. Soc. **28**, 143–166 (2002)
22. Monge, P., Contractor, N.: Theories of Communication Networks. Oxford University Press, New York (2003)
23. Meyers, L., Newman, M.E.J., Martin, M., Schrag, S.: Applying network theory to epidemics: Control measures for outbreaks of mycoplasma pneumonia. Emerging Infectious Diseases**9**(2), 204–210 (2003)
24. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. In: WWW ’004 (2004)
25. Newman, M.E.J.: The structure and function of complex networks. SIAM Rev. **45**, 167–256 (2003)
26. Rogers, E.M.: Diffusion of Innovations, 4th edn. Free Press, New York (1995)
27. Savage, S., Voelker, G.M., Paxson, G.V.V., Weaver, N.: Center for Internet Epidemiology and Defenses. http://www.ccied.org/
28. Schelling, T.: Micromotives and Macrobehavior, W.W., Norton (1978)
29. Vega-Redondo, F.: Diffusion, Search and Play in Complex Social Networks, forthcoming. Econometric Society Monograph Series (2006)
30. Lilienfeld, D.E.: Foundations of Epidemiology, 3rd edn. Oxford University Press, Oxford (1994)

Parallel Computing

► Distributed Geospatial Computing (DGC)

Parametric Model

► Hurricane Wind Fields, Multivariate Modeling

Partial Order

► Hierarchies and Level of Detail

Participation Index

► Co-location Patterns, Algorithms

Participation Ratio

► Co-location Patterns, Algorithms

Participatory Planning and GIS

CHRISTINE ROTTENBACHER
Institute of Geoinformation and Cartography,
TU Vienna, Vienna, Austria

Synonyms

Moved planning process; Negotiation; Spatial decision making of groups; Sincerity; Walking, joint; Self-referential context; Group decisions

Definition

In participatory planning processes planners have the task to organize the process of negotiation in a field of different interests and to develop a common space for acting. In participative planning processes planners care for an understanding of all steps of informing and decision making. Planners aim for sincerity, avoiding manipulation and incorrectness, showing cost, benefits and risks of decision [13,26].

Historical Background

Building a planning process from the human resources (skills, knowledge, ideas) of the participants contributes to the development of self responsibility and self initiative of the participating group. The assumption is that the shared possibilities of understanding, decision making, and acting are growing and enable a planning process of balance between an orientation-giving concept and a patchwork of decisions and actions. This orienting concept corresponds with the spatial developments which can be solved within a concrete task through an orientation in values [11,12].

Planners have to organize participative planning processes to develop a common goal in space. A participative planning process occurs in a shared constructed reality related to the concrete space. Shared reality emerges out of *interaction, understanding, decision making, negotiation, realizing, and the shared experiencing of the outcome* within groups [1,2,7,23].

The *moved planning process* is a special design of planning participation processes for empowerment of participants [21]. Usually the participation processes last for two or more years. The design of these processes is based on interdisciplinary research results about knowledge construction and decision making in groups. The research covers the changing interaction structure of the attentive subgroups of participants, the building of networks, and the decision making process of the groups.

The mutual experience of *concrete situations* create corresponding structures between participants [20]. Groups of people affected by a plan are invited to show their daily environment, the "object" of the plan. Every meeting is started with walking together through the space to be planned. After the walk follows a sitting period together to reflect the shared experiences, to find topics and tasks until the next meeting. During the walk starts an exchange about the experience and knowledge about the special place; everybody is an expert of her daily live with the place [8].

- During concrete experiences the meanings get related to the ever changing shared experiences. The previously constructed reality of individual participants often differs from the encountered reality during the walk. Differences can be pointed out and erroneous conceptions corrected. For example, plants that indicate the presence of nutrients show directly a fertilizer usage. The feedback from the visible evidence forces the participants to learn about the consequences of their actions. In the shared experience and speaking about it different realities are explained.
- The participants get familiar with the walking situation, find their style of interaction and take over roles as experts of their daily environment and show their meanings and usage of space. Then they experience movement and space.
- Body exercises, which require coordination of different body parts, strengthen the connection of neurons between many parts of the central nervous system. Even simple walking activates different neuronal networks for motor control, integrate vestibular systems, and optical and acoustical stimuli. Simple walking leads further to a parallel processing for integration of perception. Walking regularly creates an awareness of perceiving, feeling, thinking, an acting [10].
- Movement is orientated action. Joint movement leads to actions together. The theory behind different body therapy forms deals with the connection of involuntary movement and acting, and leads to insights about movement together and common acting [15]. To each personal movement-behavior belongs a shared movement-behavior [18]. This effect is exploited in groups which work with body therapeutic methods, e. g. as in authentic movement [19].

Joint walking creates experiences together, namely the common rhythm of step, the regular breathing, the physical effort, and the feeling of fatigue. Pictures are perceived through senses and attending people are encouraged to trust in their senses while moving their bodies.

Planning as an instrument for preparing decisions depends on various factors. It happens in a social system of different pressure groups. There are advices, rules, administrative and technical restrictions, right in ownership and neighborhood. There are political influences or competing projects [16].

Decision making, acting, and bearing responsibility are connected processes within the planning group. A planning group consists of person responsible, planners, and of people affected by a planning intention; at the beginning of the planning process, they form a heterogeneous group. There is no shared language, no shared way of looking at issues, no common assumptions. Every participant brings her histories and perspectives to the group. From the beginning it is necessary to look at differences and commonalities. In order to transform needs and interests into a more comprehensive understanding (which takes other needs into account), it is necessary to look at the process of shared knowledge construction in groups. Participants cannot transcend their particularity [11]. If participants make decisions appropriate to their personal context, they have to express (and need to get space for doing so) their particularity to others, and learn about the particularities of the other. This leads to a shared knowledge based on concrete situations [14]. Participants have particular knowledge that arises from experiences, also experiences in their social positions, and those social positioning influence the assumptions and interests they bring to the meeting.

Scientific Fundamentals

The Human Theory of Action [9] describes a human as an organism-environment entity. A human is embedded as a body-mind person in a social ecological environment. Experiences and knowledge are stored up in the body, and used in each concrete situation [18]. Related to these concrete situations are the possibilities of reflection and imagination. According to this theory knowledge construction and action are immediately connected.

A human is active, orientated in the future [9]. She puts her own theories and goals, and makes hypothesis about the outcome of her acting in everyday acting. These hypotheses are immediately verified through acting, and corrections of the actions. Integrated in a planning theory, it is important to consider how to integrate steps of decision making, experiencing the outcome of these decisions into the technical issues and the design of a planning process [26]. Decisions are made continuously during the planning steps.

Mutual understanding between people is possible, if the coding and decoding participants have corresponding perceiving capacities and interpretation patterns. When a planner comes to a planning group of a village a base for shared perceiving and interpretation patterns has to be found. Understanding of perceived acts becomes meaningful by the different context of social interactions.

The personal acting has a self referential meaning to give continuity and identity to the own being and acting. The "*I*" is a live system which has emotional, spiritual and cognitive abilities. This is adapting permanently the changing environment. For the internal structure it is crucial to have self awareness for these changes and anchor them to identity. A personal history is experienced and humans develop through those experiences. New competences are acquired through the contact with others.

From the base of this "*I*" it is possible to get into contact with "*you*" and with the experiences of the "*you*" [6]. Within this self referential context it is easier to understand behavior of the others as an expression of their "*I*".

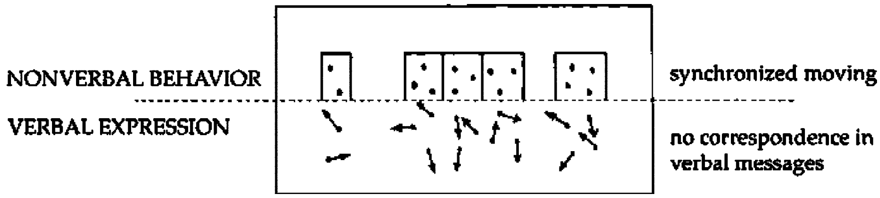
At a meeting and a joint walk this concrete context is strengthened. People find a common rhythm and breath, and have joint experiences. The immediate behavior evokes primarily patterns of existence and less patterns of thinking. New experiences, understanding, and knowledge are shared [14]. The participants experience themselves mutually within every new meeting and the structure of interaction in the concrete situation is new defined.

Participants develop new roles and test them in new behaving patterns and acting. This is the condition that they are able to perceive new contents and information, and integrate it into their personal experience, knowledge, and acting. This flexible interaction enables to anticipate and imagine the future [17].

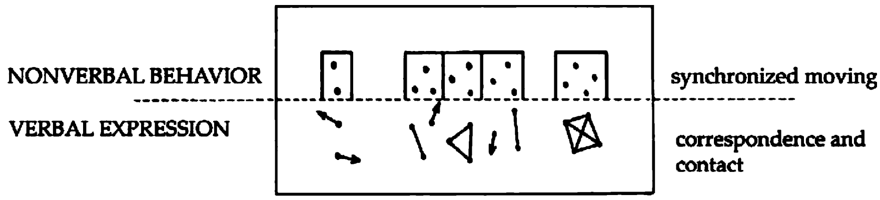
During walking the meaning of symbols are related to the ever changing joint experiences. Our understandings of symbols and behavior are exchanged. Not only are meanings expressed, but humans become more aware of what others think of them [8]. Through communication, humans look at themselves through the perspective of their partners, and take on their role. Mutual expectations have an effect on communication. When a person is walking, the location is present as a context for talking. There is one base of experiencing an understanding reality.

The research points out that concrete experienced correspondence in a group links participants and leads to grounded knowledge construction and decision making. Empirical evidence shows that networks between participants are strengthened because of the common experience in concrete situations [14,15,18]. The usual group-, and decision structures are enlarged and new ones rebuilt by experiencing movement and living space; and a shared reality grows avoiding a bargaining of interests. Walking in concrete situations contributes to the decision process in groups because it strengthens the concrete experience.

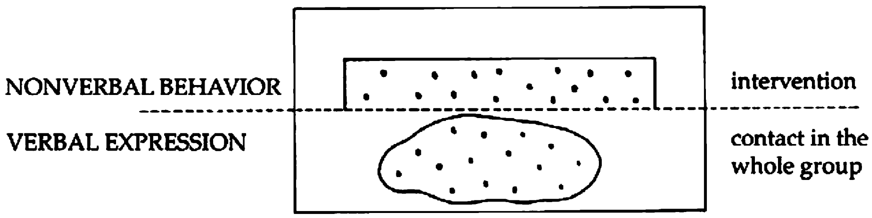
Participants form subgroups reflecting the social structure of the village. Walking breaks up subgroups and aids interactions among all participants [3]. The emotional correspondence is increasing in these changing subgroups. After the start of walking the different subgroups reach



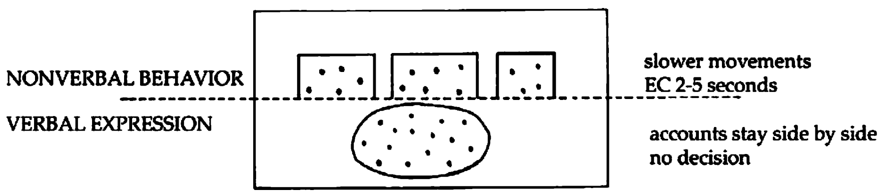
Participatory Planning and GIS, Figure 1 Correspondence in Movement



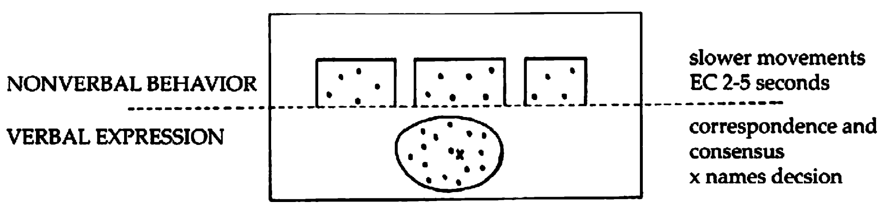
Participatory Planning and GIS, Figure 2 Correspondence in Movement and Contact in Verbal Expressions



Participatory Planning and GIS, Figure 3 Correspondence and Contact in the Whole Group



Participatory Planning and GIS, Figure 4 Correspondence without Decision in the Whole Group



Participatory Planning and GIS, Figure 5 Correspondence with Decision in the Whole Group

a synchronized moving which is observable [4,5]. In comparison with verbal expressions it is ascertainable that at that time there was mostly no correspondence in verbal expressions as shown in Fig. 1.

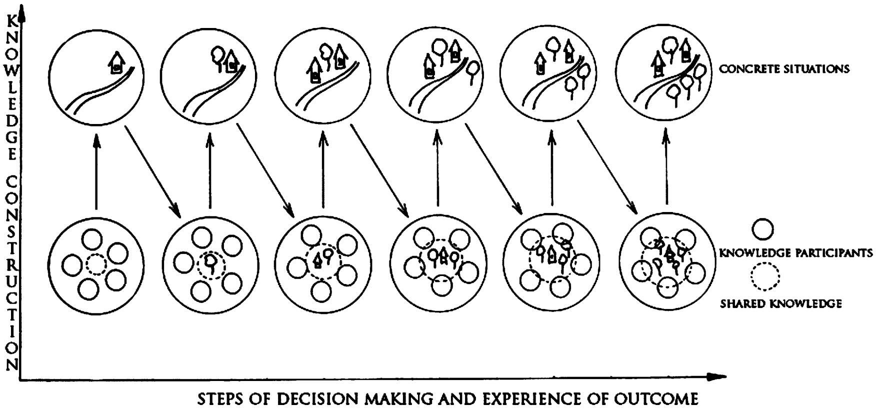
Figure 2 shows the beginning verbal correspondence and contact in the subgroups.

Figure 3 shows an event from the whole group. The subgroups open and experience together a concrete situation, exchange state, attitudes, and meanings. After this "intervention" the participants keep together, and build new subgroups within the whole group relating to the topics they found together.

Figures 4 and 5 show generalizations of events during the sitting period within the whole group. An emotional correspondence in nonverbal behavior is observable. Typically the movements get slower, and participants turn towards each other. Figure 4 describes the situation when participants stay side by side without discussion and without a decision.

Figure 5 describes the comparison of nonverbal correspondence with verbal expression. Here a person named a decision and all participants agreed. These decisions are then realized quickly.

During the sitting phase participants narrate in the first person. They start "naming" what they experienced and perceived, and they "name" their joint expectations. The participants make decisions, recognize jointly tasks, and assign duties. Small successes advance the process of decisions and acting, and create an identity in the group. This interaction base is strengthened with every new meeting.



Participatory Planning and GIS, Figure 6 Shared Knowledge Construction

Future Directions

The objective of participative planning processes is to develop a common goal in physical space. To develop a common goal it is necessary to get knowledge about social and physical usage of space. Knowledge construction, decision making, acting, and the bearing of responsibility are immediately connected within the planning steps of the planning group. The outcome of participative planning processes should be a special form of collaboration organizing social and physical usage of space. Therefore a shared knowledge construction is one important preconception for successful public participation [24].

Movement together through the space to be planned increases an *inter corporal* existence of the group. This inter corporal existence is the basic human experience of relationship, and contains all information of experiences and knowledge, and influences feeling, thinking, and acting patterns. In this inter corporal existence participants can feel, see, and interpret the actions and intentions of other participants.

Imagination and understanding emerges from the embodied experiences [25]. Human bodily movement and interaction integrate recurring patterns and develop new ones. It is possible to integrate information and transform it into knowledge in a mutual experiencing and understanding. Joint movement bring up joint experiences.

This concept is the backdrop for the assumptions about understanding, decision making, and knowledge construction processes among participating persons within a planning process. They integrate information and transform it into knowledge in a mutual understanding. Experiencing actions together leads to a shared knowledge construction. The knowledge the participants just brought with them, their constructions, their feeling and thinking patterns, remain. The concrete experience enlarges the shared knowledge [14]. The participants have to experience the concrete outcome of first decisions to strengthen the base of common action. Structures of social interaction are opened and renew the base of contact. The participants experience themselves mutually within every new meeting; they experience the concrete situation, make a common decision about this situation, and experience the first outcome of the decision.

Further the participants develop slightly changed roles and test them in new behaving patterns. This is the pre-condition: they are able to perceive new contents and information, and integrate it into their personal experience, knowledge, and acting. This flexible interaction enables to anticipate and imagine the future, to make decisions, to act, and to experience the outcome step by step.

How participants get more and more linked is shown in the next two figures.

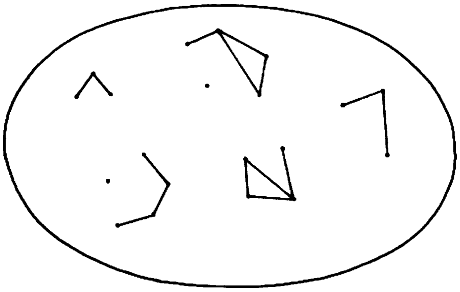
Figure 7 shows the usual structure of the whole group. The subgroups build by the participants relate to the social structure of the village.

After a meeting and after experiencing concrete situations participants are better linked [22].

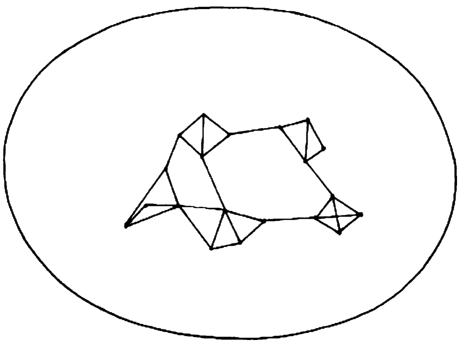
Key Applications

For planning issues GIS should support the context of *knowledge construction* of planners and participants to create meaning out of collaborative extraction of information, and to contribute to *group decisions* that use geospatial information.

To achieve effective group work with geospatial information it is essential to ask how the common experience of concrete situations can be combined with mediated decision making within a planning context. This suggests 1) that in virtual systems attention must be paid to the connection of the virtual situation to concrete previous experience of participants so that they are able to integrate new contents and information into their personal and shared experience, knowledge, acting, and experiencing of the outcome of acting, and 2) the interaction among par-



Participatory Planning and GIS, Figure 7 Initial Group Structure



Participatory Planning and GIS, Figure 8 Connection of subgroups

ticipants in virtual communities should be structured to achieve changing subgroups for building networks relating to changing tasks.

Cross References

► Decision-Making Effectiveness with GIS
► Geocollaboration

Recommended Reading

1. Argyle, M.: Bodyly Communication. Methuen & Co., London (1975)
2. Axelrod, R.: The Evolution of Cooperation. Basic Books, Princeton (1997)
3. Bakeman, R., Gottman, J.M.: Observing Interaction. Cambridge University Press, Cambridge, New York (1997)
4. Bales, R.F.: Interaction process analysis. Cambridge/Mass., Cambridge, New York (1951)
5. Bales, R.F.: Die Interaktionsanalyse. Ein Beobachtungsverfahren zur Untersuchung kleiner Gruppen. In: König, R. (ed.) Praktische Sozialforschung II. Beobachtung und Experiment in der Sozialforschung, pp. 148–167. Kiepenheuer & Witsch, Köln, Berlin (1962)
6. Buber, M.: Ich und Du. Reclam jun. GmbH & Co., Stuttgart (1995)
7. Damasio, A.: The Feeling of What Happens. First Harvest edition, San Diego, New York, London (1999)
8. Goffman, E.: Dramaturgie des Rollenhandelns. In: Retter, H.: Studienbuch Pädagogische Kommunikation. Klinkhardt, Bad Heilbrunn (2002)
9. Goldstein, K.: The Organism. Aholistic Approachto Biology Derived from Pathological Data in Man. Zone Books, New York (2000, 1934)
10. Hannaford, C.: Smart Moves: Why Learning is not all in your Head. Great Ocean Publishers, Arlington, VA (1995)
11. Haraway. D.: Situated Knowledges: The Science Question in Feminism and the Priviledge of Partial Perspective. In: Simians, Cyborgs, and Women. Routledge, New York, (1991)
12. Hochschild, J.: Where You Stand Depends on What You See: Connections among Values, Perceptions of Fact, and Prescriptions. In: Kuklinski, J. (ed.) Citizens and Politics: Perspectives from Political Psychology. Cambridge University Press, Cambridge (2000)
13. Jacobs, J.: The Death and Life of Great American Cities. Vintage Books, New York (1961)
14. Johnson, M.: The Body in the Mind. Chicago Press, Chicago (1987)
15. Lewin, K.: Field Theory in Social Science. Selected Theoretical Papers. D. Cartwright. Harper&Row, New York (1951)
16. Mansbridge, J.: Beyond Adversary Democracy. Basic Books, New York (1980)
17. Maturana, H.R.: Biologie der Realität. Suhrkamp, Frankfurt am Main (1998)
18. Merleau-Ponty, M.: Phänomenologie der Wahrnehmung. Walter De Gruyter & Co., Berlin (1966)
19. Pallaro, P. (ed.): Authentic Movement. Jessica Kingsley Publishers, London (1999)
20. Rottenbacher, C.: Presence in the Planning Process. GEOS 2004. Brasilien, (2004)
21. Rottenbacher, C.: Motion Increeases Emotional Correspondence in Geocollaboration. GIScience 2004. Baltimore, (2004)
22. Rottenbacher, C.: Emotional Correspondence Links. CORP 2005 (2005)
23. Rottenbacher, C.: Shared Knowledge Construction in Heterogeneous Groups. CORP 2006 (2006)
24. Smith, M.K.: Kurt Lewin, Groups, Experiential Learning and Action Research. The Encyclopedia of Informal Education. http://www.infed.org/thinkers/et-lewin.html (2001)
25. Varela, F.J., Thompson, E., Rosch, E.: The Embodied Mind. MIT Press, Cambridge Massachusetts (1997)
26. Young, M.I.: Inclusion and Democracy. Oxford Political Theory. Oxford University Press, Oxford, New York (2000)

Partitioning

► Geodemographic Segmentation

Path, Space-Time

► Time Geography

Pattern, Encounter

► Movement Patterns in Spatio-temporal Data

Pattern, Flock

► Movement Patterns in Spatio-temporal Data

Pattern, Leadership

► Movement Patterns in Spatio-temporal Data

Pattern, Moving Cluster

► Movement Patterns in Spatio-temporal Data

Pattern, Periodic

► Movement Patterns in Spatio-temporal Data

Pattern Recognition in Spatial Data

► Geographic Knowledge Discovery

Patterns

► Data Analysis, Spatial

Patterns, Complex

SANJAY CHAWLA
School of Information Technologies, The University of Sydney, Sydney, NSW, Australia

Synonyms

Correlated; Negatively-correlated; Spatial association; Co-location; Frequent itemset mining

Definition

Complex spatial relationships capture self, positive, negative and mixed relationships between spatial entities. These relationships abstract the commonly occurring relationships in ecology, cosmology and other disciplines where spatial referencing plays an important role. Table 1 enumerates the different types of complex relationships using the example of elliptical and spiral galaxy types from the field of cosmology. More formally, the definition of complex relationships is predicated on the concept of colocation.

Definitions 1 *(Co-location) Two spatial objects are said to co-locate if the Euclidean distance between the objects is less than or equal to the user-specified neighborhood distance d.*

Definitions 2 *(Positive) A positive relationship in spatial data is a set of features that co-locate at a ratio greater than some predefined threshold. In spatial data, the confidence of a positive relationship* $A \to B$ *is given by the fraction of unique Bs that co-occur in a clique containing the feature A.*

Definitions 3 *(Negative) A negative relationship in spatial data is defined as where a feature is absent from a given co-location at a ratio greater than a predefined threshold. Negative relationships are often denoted by "−".*

Definitions 4 *(Self-Co-location) A feature is defined as self-co-locating in spatial data if the items representing that feature co-locate with each other at a ratio greater than some predefined threshold. A Self-Co-location is denoted by "+".*

Definitions 5 *(Self-Exclusion) A feature is defined as self-excluding in spatial data if the items representing that feature co-locate with each other at a ratio less than some predefined threshold.*

Definitions 6 *(Complex) A complex relationship in spatial data is any relationship containing the properties of two or more of the other relationships.*

Historical Background

The study of spatial point processes is a core topic of research in the spatial statistics community [2,3,6]. The breakthrough in the data mining community is due to a series of papers by Shekhar et al. [8] and Huang et al. [4]. The two papers introduced a series of new *measures* which made it possible to efficiently discover colocation relationships in large spatial data sets. Furthermore, the *measures* introduced were closely related to the cross-K function [2]. This made it possible to *mine* for spatial relationships as opposed to *test* for them, which is the usual practice in Statistics.

Scientific Fundamentals

The methodology of discovering complex spatial relationships is based on a foundational data mining framework known as *frequent itemset mining*. The basic idea is as follows. Suppose there are *n* binary variables (also known as items or features) and the objective is to discover which elements of the power set of the *n* variables are correlated. A brute-force approach is computationally infeasible because the size of the search space is exponential in the number of variables. Instead, the elements of the power set can be examined in a leveled fashion and pruned as the power set lattice is being examined. The basic insight (known as the Apriori or anti-monotonic Property) is that

Patterns, Complex, Table 1 Types of Complex Spatial Relationships [5]

Relationship	Notation	Description	Example
Positive	$A \to B$	Presence of B in the neighborhood of A	Sa type Spiral Galaxies → Sb type Spiral Galaxies
Negative	$A \to -B$	Absence of B in the neighborhood of A	Elliptic galaxies tend to exclude spiral galaxies. E → −S
Self-Co-location	$A \to A+$	Presence of many instances of the same feature in a given neighborhood	Elliptic galaxies tend to cluster more strongly. E → E+
Self-Exclusion	$A \to -A+$	Absence of many instances of the same feature in a given neighborhood	Two or more of the same type of spiral galaxies are rarely found in the same neighborhood. Sa → −Sa+
Complex	$A+ \to -C,B$	combination of two or more of the above relationships	Clusters of elliptic galaxies tend to exclude other types of galaxies. E+ → −S

If a set of variables is not interesting, then neither are its supersets. This observation can be used to prune the power set lattice of variables in a computationally efficient manner.

The notion of *interestingness* is crucial in the frequent itemset mining framework. For example, the traditional measure of correlation does not satisfy the Apriori Property, but the simple COUNT function (known as the support) does. Huang et al. [4] have introduced the Maximal Participation Index (maxPI), which possesses a weak form of the Apriori Property and can be used to discover rare but interesting spatial relationships. Later, Arunasalam et al. [1] have formally and empirically shown how maxPI can be used to mine for complex spatial relationships. We briefly elaborate on the maxPI measure.

(Participation ratio) Given a co-location pattern L and a feature $f \in L$, the participation ratio of f, $\mathrm{pr}(L, f)$, can be defined as the support of L divided by the support of f. For example, in Fig. 1, the support of $\{A, B, C\}$ is 2 and the support of C is 6, so $\mathrm{pr}(\{A,B,C\},C) = 2/6$.

(Maximal Participation Index) Given a co-location pattern L, the maximal participation index of L, $\mathrm{maxPI}(L)$ can be defined as the maximal participation ratio of all the features in L, i. e., $\mathrm{maxPI}(L) = \max_{f \in L}\{\mathrm{pr}(L,f)\}$. For example, in Fig. 1,

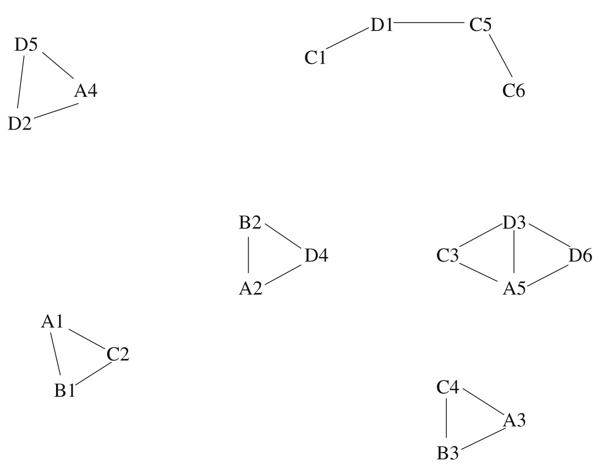
$$\mathrm{maxPI}(\{A, B, C\}) = \max\left(\frac{2}{5}, \frac{2}{3}, \frac{2}{6}\right) = \frac{2}{3}\,.$$

A high maximal participation index indicates that at least one spatial feature (which we call the *maxfeature*) strongly implies the pattern. By using maxPI, rules with low frequency but high confidence can be found, which would otherwise be pruned by a support threshold.

Maximal participation index is not anti-monotonic with respect to the pattern containment relations. For example, in Fig. 1, $\mathrm{maxPI}(\{A,C\}) = 3/5 < \mathrm{maxPI}(\{A,B,C\}) = 2/3$. Interestingly, the maximal participation index does have the following weak monotonic property:

If P is a k-co-location pattern, then there exists at most one $(k-1)$ subpatterns P' of P such that $\mathrm{maxPI}(P') < \mathrm{maxPI}(P)$.

Complex relationships are not restricted to mining complex rules. Complex relationships can be used to provide stronger definitions and more accurate significance testing for simple relationships. In terms of confidence, the significance of a rule is given by the extent to which the observed confidence of a rule differs from the expected confidence given by a random distribution. Given a set of confident rules, the significance of these rules will depend on the relative size of the space from which they were taken. For example, Munro et al. [5] have shown that



Patterns, Complex, Figure 1 An Example of spatial co-location patterns

Patterns, Complex, Table 2 Cliques in Fig. 1

No	Clique
i	C_1, D_1
ii	C_5, D_1
iii	C_5, C_6
iv	A_4, D_2, D_5
v	A_5, C_3, D_3
vi	A_5, D_3, D_6
vii	A_1, B_1, C_2
viii	A_2, B_2, D_4
ix	A_3, B_3, C_4

The significance of a confident rule of the form $A \to B$ is independent of the self-co-location/exclusion of A, but is dependent on the self-co-location/exclusion of B.

Confidence is a measure of conditional probability. The confidence of $A \to B$ is $P(B|A)$. Thus, the probability of finding instances of B in a clique where A already exists is measured by the propensity of B's to appear in the clique. This is the same if instances of A appear in one (self-colocation) or every clique (exclusion).

Key Applications

Traditional statistical techniques are designed for hypothesis testing: *Is the spatial relationship between two features significant?* In data mining, where large data sets with a multitude of features are the norm, the question of hypothesis generation is perhaps also interesting: *Find all multiple combinations of features such that the spatial relationship between these features is potentially significant*. Accurate and computationally efficient methods of discovering spatial relationships will help domain experts in diverse domains such as anthropology, cosmology, ecology, epidemiology, and geophysics. Additionally, many other disciplines unlock and discover new relationships and candidate theories in their respective fields.

Future Directions

One of the key computational challenges is to scale spatial collocation algorithms to handle an increasing number of spatial features. In traditional frequent mining, single features (items) can be pruned by themselves. However, because of the need to capture spatial relationships, single features cannot be pruned on their own. Thus, all pairs of features have to be initially computed leading to at least quadratic complexity in the number of features.

Another challenge is to extend the discovery the approach of mining for complex relationships to a spatio-temporal setting. As noted in Schabenberger et al. [7], observed point patterns are a snapshot of evolving patterns. For example, naturally generating oak trees initially tend to be clustered, then seem to be randomly distributed and finally tend to be arranged in a regular pattern as they compete for more space. Designing data mining techniques which can capture evolving trends in a spatio-temporal setting provides an exciting opportunity for future research.

Recommended Reading

1. Arunasalam, B., Chawla, S., Sun, P.: Striking two birds with one stone: Simultaneous mining of positive and negative spatial patterns. In: Proceedings of 2005 SIAM International Conference on Data Mining SDM05, to appear (2005)
2. Noel, A., Cressie, C.: Statistics for spatial Data. John Wiley and Sons, New York (1993)
3. Diggle, P.J.: Statistical Analysis of Point Processes. Chapman and Hall, New York (1983)
4. Huang, Y., Xiong, H., Shekhar, S., Pei, J.: Mining confident co-location rules without a support threshold. In: Proceedings of the 18th ACM Symposium on Applied Computing ACM SAC (2003)
5. Munro, R., Chawla, S., Sun, P.: Complex spatial relationships. In: Proceedings of the 3rd IEEE International Conference on Data Mining, ICDM 2003, pp. 227–234. IEEE Computer Society (2003)
6. Ripley, B.D.: Spatial Statistics. John Wiley and Sons, New York (1981)
7. Schabenberger, O., Gotway, C.: Statistical Methods for Spatial Data Analysis. Chapman and Hall, (2005)
8. Shekhar, S., Huang, Y.: Discovering spatial co-location patterns: a summary of results. In: Proceedings of the 7th International Symposium on Spatial and Temporal Databases SSTD01 (2001)

Patterns in Spatio-temporal Data

HUI YANG[1], SRINIVASAN PARTHASARATHY[2]

[1] Department of Computer Science and Engineering, San Francisco State University, San Francisco, CA, USA

[2] Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, USA

Synonyms

Evolving spatial patterns; Spatio-temporal association patterns; Spatio-temporal object association

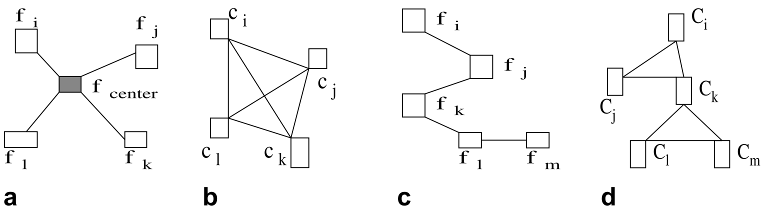
P

Definition

Spatio-temporal data refer to data that are both spatial and time-varying in nature, for instance, the data concerning traffic flows on a highway during rush hours. Spatio-temporal data are also being abundantly produced in many scientific domains. Examples include the datasets in computational fluid dynamics that describe the evolutionary behavior of vortices in fluid flows, and the datasets in bioinformatics that study the folding pathways of proteins from an initially string-like 3D structure to their respective native 3D structure.

One important issue in analyzing spatio-temporal data is to characterize the spatial relationship among spatial entities and, more importantly, to define how such a relationship evolves or changes over time. In the traffic flow example, one might be interested in identifying and monitoring the automobiles that are following one another far too close. Such an issue is often summarized as finding interesting spatio-temporal patterns.

A spatio-temporal pattern characterizes the spatial relationship among a collection of spatial entities and the evo-



Patterns in Spatio-temporal Data, Figure 1 Examples of spatial association patterns. **a** Star. **b** Clique. **c** Sequence. **d** minLink=2

lutionary behavior of such a relationship over time. As an example, Fig. 1 illustrates four types of spatial patterns, corresponding to four different types of spatial association. In this figure, each rectangle represents a spatial entity, and an edge indicates that the two involved entities hold a certain spatial relationship. For instance, an edge can mean that the Euclidean distance of the two entities is within a specified threshold. It can also mean that one entity is located to the left of the other. Or it can mean both of the above relations hold between the two entities. Assume that a collection of spatial entities $E = (e_0, e_1, \ldots, e_k)$ formed a star-like pattern (Fig. 1a) at time t_1 and continued in this fashion until time t_2. One can employ a spatio-temporal pattern in the form of $(Star, E, t_1, t_2)$ to effectively model such an evolving process.

Due to the following reasons, spatio-temporal patterns are often multifaceted. Furthermore, the spatio-temporal characteristics captured by such patterns often vary from application to application [6,9,10]:

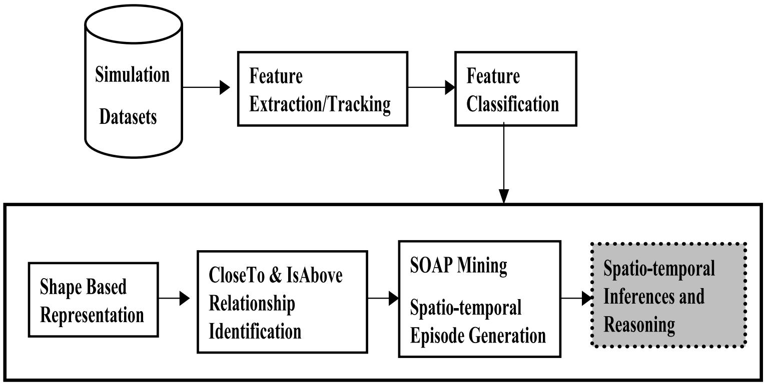
- Diversity of spatial relationship. For any pair of spatial entities, there exist a variety of spatial relations between them, such as directional relation, distance-based relation, and topological relation. Which of these relations should be captured in a spatio-temporal pattern is often specific to individual applications.
- Complexity of temporal relationship. For instance, there exist 13 possible relations between two time intervals [1]. Again, it is often governed by the applications to decide what relations should be considered in the spatio-temporal patterns.
- Representation of spatial entities: points or geometric objects?
- Varying application-specific requirements. For instance, one application might require one to capture how the distances between entities change in time, whereas another application might be interested in investigating both the distance and relative directional arrangement between entities.

Note that evolving spatial clusters—collections of spatial entities that are similar to one another (e. g., entities within the same vicinity)—are another type of spatio-temporal patterns. The main difference between evolving spatial clusters and the above-described spatio-temporal patterns resides in the number of involved spatial entities. A spatial cluster often consists of much more spatial entities than a spatio-temporal pattern. Additionally, spatio-temporal patterns are more versatile in the sense that a variety of spatial and temporal relations can be considered simultaneously as needed, whereas spatial clusters are often concerned about only the distance-based relationship among entities.

Historical Background

The history of spatio-temporal association patterns is closely related to that of spatial association patterns, since the former is often derived by incorporating the temporal dimension into the latter. Spatial association patterns were first studied by Koperski and Han [4]. This early work focuses on extracting patterns specified in advance. Following this work, a considerable amount of work was conducted to detect spatial clusters [2]. Such clusters mainly captured the spatial proximity among entities. Driven by the widespread location-based services at the turn of the century, researchers started to take a special interest in identifying spatial patterns that involve a smaller set of entities within a confined spatial neighborhood [6]. Such patterns were later termed as spatial collocation patterns [3]. For instance, the collocation pattern *(weather, airline schedule, Starbucks coffee shops)* captures the phenomenon that the customers at Starbucks coffee shops tend to request weather information and airline schedules together through cellular phone. However, the research work up to this point often simplified spatial entities to point objects and mainly considered the Euclidean distance between objects. Recently, several studies were carried out to overcome such limitations [9]. In these studies, spatial entities are represented as geometric objects of different shape and size. In addition, the spatio-temporal patterns are capable to capture multiple spatial relations (e. g., both distance-based and directional relation). Consequently, the term spatial or spatio-temporal object association patterns were coined to emphasize such facts [10].

Another prominent development of spatio-temporal patterns analysis is that it has found more and more applications in scientific domains, such as astronomy, meteorology, biochemistry, and bioinformatics. This is in contrast to its earlier application mainly in geographic information systems.



Patterns in Spatio-temporal Data, Figure 2
A Generalized Framework for Analyzing Spatio-temporal Scientific Data

Scientific Fundamentals

The process of identifying spatio-temporal patterns can be decomposed into three main phases. The first phase is data preprocessing. Main tasks in this phase include the following: (1) Determine the representation scheme of spatial entities: points or geometric objects? If it is the latter case, what geometric properties and domain specific attributes need to be considered? (2) Concretize the spatio-temporal patterns: what spatial and temporal relations should the patterns be modeling? (3) Identify and define the measurements that measure the "interestingness" of a pattern. For instance, *support* and *prevalence* have been proposed by Yang *et al.* to characterize the significance of a pattern [10]. The second phase is to efficiently and effectively discover interesting spatio-temporal patterns. One main challenge is to achieve good scalability and performance in the presence of a large volume of data, which are often in the range of gigabytes and even terabytes. Efficient data structures and optimization strategies are often employed towards improving scalability and performance. The third and final phase is to evaluate the identified spatio-temporal patterns and put them into use. The nature and implementation of this phase is often application-specific.
In scientific domains, the discovery of spatio-temporal patterns often brings up new challenges. For instance, to discover spatio-temporal patterns of vortices in fluid flows, one needs to first detect and extract vortical objects at different time. This task by itself is still under intensive study currently. Readers are referred to [10] for more details on a generalized framework for analyzing scientific spatio-temporal data. This framework is illustrated in Fig. 2.

Key Applications

Spatio-temporal association patterns have been used to address various issues in many domains. Below is a list of representative applications from different domains.

Traffic Management

Spatio-temporal association patterns can be used to identify and predict potential accidents by modeling automobiles within dangerous distance. Such patterns can also be used to redirect traffic flows, thereby avoiding potential traffic jam.

Behavior Tracking in Security Surveillance Systems

Surveillance systems track and record the behavior of human subjects aiming at identifying suspicious behaviors. One can use spatio-temporal patterns to model such behaviors by associating a person's movement with objects in the surrounding area.

Astronomy

In astronomy, spatio-temporal patterns can be used to capture the evolution of interactions among astronomical objects in the vicinity by exploring the data accumulated in the past.

Transmissible Disease Control

To control and predict the spreading rate of transmissible diseases (e. g., SARS), one critical issue is to have a clear notion of how people in the infected areas regularly relate to each other and with people in the disease-free areas. Spatio-temporal association patterns can be applied to model such people-people interactions.

Computational Molecular Dynamics: Interaction and Evolution of Defects in Materials

It has been observed that multiple defects in materials often interact with each other. Such interactions eventually might lead to undesirable results, such as the amalgamation of small defects and the breakdown of large defects. Again, such behavior can be modeled and captured by identifying spatio-temporal association patterns of defects.

Computation Fluid Dynamics: Characterizing Vortical Flows

Vortices—swirling regions around a common center—in vortical flows can often produce undesirable effects, especially when such vortices interact with one another. For instance, vortices in the air flows surrounding an airplane can lead to audible noise and strong vibration. Therefore, designers often resort to computer simulations to study vortical flows around a certain model. Here one can use spatio-temporal patterns to characterize the evolving behavior of vortices at different locations of the model under study.

Bioinformatics: Protein Folding Trajectories Analysis

A protein folding trajectory describes the folding path of a protein from an initially string-like structure to its final native and often complex structure. Along this path, amino acids, the building blocks of a protein, interact with one another. Such interactions often result in a variety of folding events, such as nucleation and secondary structure formation. It has been demonstrated that spatio-temporal association patterns could be applied to address several issues: (1) summarizing a folding trajectory; (2) detecting and ordering folding events along a trajectory; and (3) identifying a consensus partial folding pathway across different trajectories of a protein [11].

Future Directions

Discovering interesting and meaningful spatio-temporal association patterns is still a relatively new problem. Below are several potential research focuses related to this problem: (1) design scalable algorithms that can handle large volume of spatio-temporal datasets. Candidate solutions include the following: integrating efficient indexing schemes in the process and developing parallel or distributed algorithms; (2) implement effective approaches to incorporate domain-specific knowledge in the pattern discovering process; (3) utilize visualization techniques to facilitate an easier verification and a better understanding of the discovered spatio-temporal patterns; and (4) implement generalized software systems to discover spatio-temporal patterns similar application domains.

Recommended Reading

1. Allen, J.F.: Maintaining knowledge about temporal intervals. Commun. ACM. **26**(11), 832–843 (1983)
2. Ester, M., Kriegel, H.P., Sander, J.: Algorithms and applications for spatial data mining. Geographic Data Mining and Knowledge Discovery, Research Monographs. In: GIS Chapter 7 (2001)
3. Huang, Y., Xiong, H., Shekhar, S., Pei, J.: Mining confident co-location rules without a support threshold. In: SAC 03: Proceedings of the 2003 ACM symposium on applied computing, pp. 497–501. ACM Press (2003)
4. Koperski, K., Han, J.: Discovery of spatial association rules in geographic information databases. In SSD 95: Proceedings of the 4th International Symposium on Advances in Spatial Databases, pp. 47–66. Springer-Verlag (1995)
5. Mokbel, M.F., Ghanem, T.M., Aref, W.G.: Spatio-temporal access methods. Technical report, Department of Computer Sciences, Purdue University
6. Morimoto, Y.: Mining frequent neighboring class sets in spatial databases. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 353–358. ACM Press (2001)
7. Neill, D.B., Moore, A.W., Sabhnani, M., Daniel, K.: Detection of emerging space-time clusters. In: Proceedings of SIGKDD 2005, pp. 218–227 (2005)
8. Rao, C.R., Suryawanshi, S.: Statistical analysis of shape of objects based on landmark data. Proc Natl Acad Sci U S A. **93**(22), 12132–12136 (1996)
9. Xiong, H., Shekhar, S., Huang, Y., Kumar, V., Ma, X., Yoo, J.S.: A framework for discovering co-location patterns in data sets with extended spatial objects. SIAM Intl. Conf. on Data Mining (SDM), April 2004
10. Yang, H., Parthasarathy, S., Mehta, S.: A generalized framework for mining spatio-temporal patterns in scientific data. In KDD 2005: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pp. 716–721. ACM Press, New York, NY, USA (2005)
11. Yang, H., Parthasarathy, S., Ucar, D.: A spatio-temporal mining approach towards summarizing and analyzing protein folding trajectories. Algorithms Mol. Biol. **2**(3) (2007)

Peano Curve

► Indexing of Moving Objects, B^x-Tree

Peer Data Management

► Database Schema Integration

Peer to Peer

► Cloaking Algorithms for Location Privacy

Peer-to-Peer Caching for Spatial On-Line Analytical Processing

► Olap Results, Distributed Caching

Peer-Tree (Spatial Index)

► Data Collection, Reliable Real-Time

Perceptory Pictograms

► Modeling with Pictogrammic Languages

Personalization

► Geospatial Semantic Web: Personalisation
► User Interfaces and Adaptive Maps

Personalized Maps

► Mobile Usage and Adaptive Visualization

Personalized Visualization

► Mobile Usage and Adaptive Visualization

Phantom Update Protection

► Concurrency Control for Spatial Access Method

Phenomenon Spatial Field

► Geosensor Networks, Estimating Continuous Phenonena

Photogrammetric Applications

THOMAS LUHMANN
Institute for Applied Photogrammetry
and Geoinformatics, Oldenburg, Germany

Synonyms

Photogrammetry; Application; Aerial; Close range; Data acquisition; 3D city models

Definition

The numerous application areas for photogrammetry can be categorized into two major groups: topographic applications based on aerial and satellite imagery, and close-range applications.

Historical Background

Photogrammetry was invented between 1850 and 1860 independently by the French Laussedat and the German Meydenbauer. Aerial photogrammetry became a focus with the development of aircraft and fast film material. It was further developed by the coming of color film and, in the second half of the twentieth century, by satellite imaging systems. The photogrammetric industry developed analog optical and mechanical stereo plotting devices and image rectifiers to meet the practical demands. Since the 1960s, computer-based analytical processing methods have been used intensively. Consequently, digital photogrammetry became state of the art in the 1990s, when digital imagery and digital imaging devices came into use. In close-range photogrammetry, the main fields of application were addressing architecture and cultural heritage, accompanied by a variety of very specialist applications and solutions. With the development of self-calibrating bundle adjustment programs around 1980, industrial photogrammetry was strongly used mainly for large scale metrology, e. g., for antennae or in the aerospace industry. Again, the development of high-resolution digital cameras pushed the technique further ahead. Nowadays, photogrammetry is an accepted and widely used tool in many industrial, medical, and engineering tasks.

Scientific Fundamentals

Please refer to the entries on Mathematical Concepts of Photogrammetry, Photogrammetric Products, and Photogrammetric Sensors.

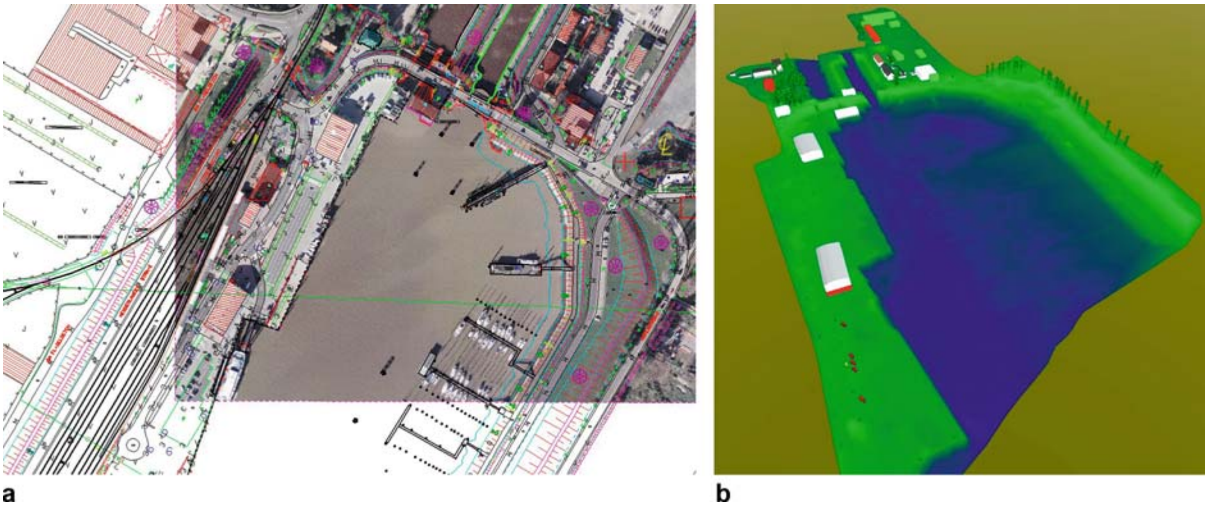
P

Key Applications

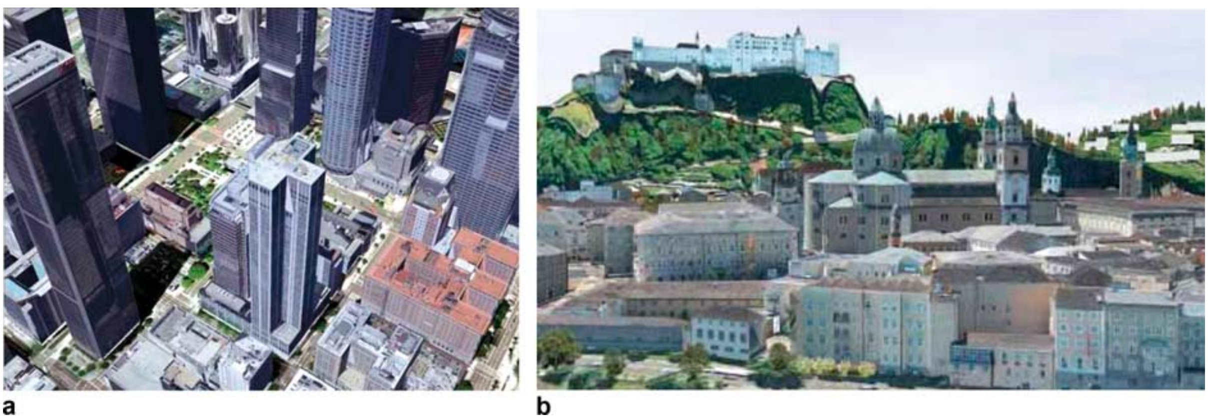
Aerial Applications

Applications in aerial photogrammetry can be characterized by often similar imaging configurations, i. e., equal or similar cameras, nadir imagery, large imaging distances (flying height) and image scales usually between 1:2,000 and 1:30,000. The most important products of aerial photogrammetry are orthophotos, 3-D terrain and city models, and vector data usually used as input for geographic information systems (GIS).
The following examples show a small spectrum of the applications in aerial photogrammetry.

GIS Data Acquisition Photogrammetric data generation for GIS purposes is the most important application of aerial photogrammetry. An example of photogrammetric data acquisition and modeling for a GIS application is shown in Fig. 1. The task was the extraction of data about the harbor of the German town Emden in order to provide 2-D and 3-D information for an internet-based information system. Besides aerial photographs, additional data sources such as cadastral maps, sonar depth measurements and terrestri-



Photogrammetric Applications, Figure 1 Photogrammetric data extraction and modeling for a harbor information system. **a** Superimposition of **geographic information systems** (GIS) data and aerial image. **b** 3-D depth model of port basin

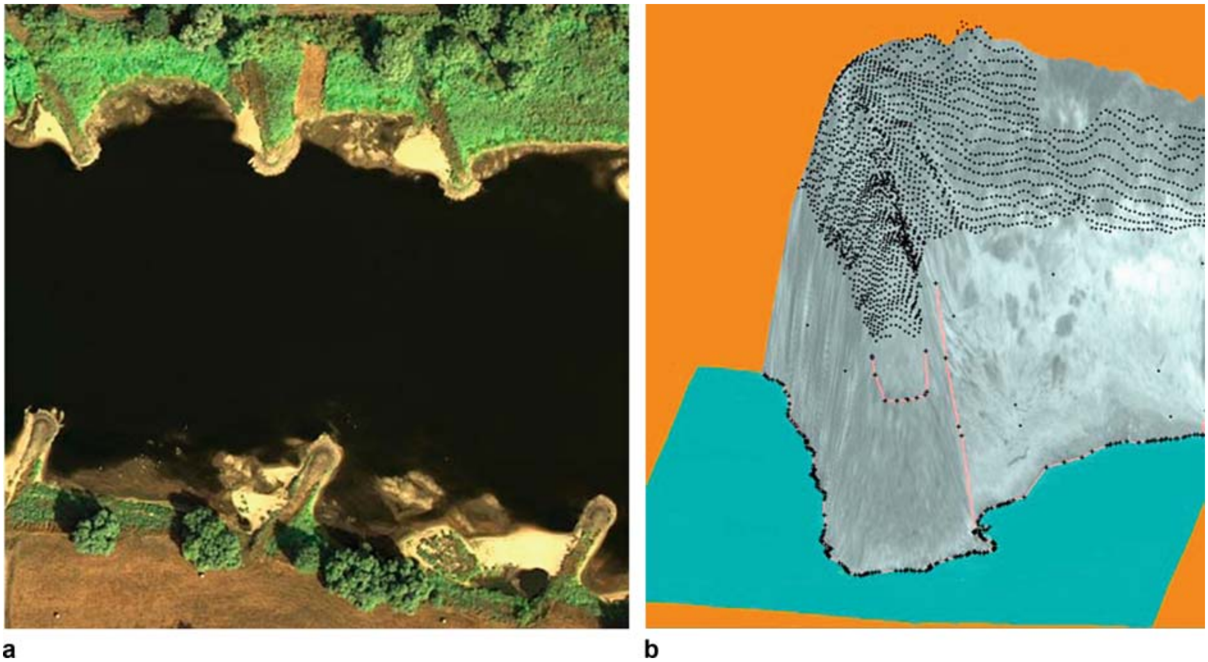


Photogrammetric Applications, Figure 2 Examples of 3-D city models from Los Angeles and Salzburg (CyberCity)

al images have been used. Photogrammetric data compilation is performed using a digital stereo workstation.

3-D City Models Aerial images are the most important data source for measuring 3-D city models. Image information is used to extract 3-D points and topological information (wire frame models). In addition, the operator can identify different types of buildings and other objects in order to classify the scene in terms of database attributes. 3-D city models are increasingly used for touristic purposes, urban planning, real estate management and emission monitoring and prediction. Figure 2 shows examples of city models that have been extracted from aerial imagery interactively. The facades of the buildings can be textured from terrestrial images. For visualization of the resulting huge amount of data, specialized visualization software and data structures are employed.

Flood Monitoring Monitoring of rivers is becoming increasingly important for environmental protection and flood disaster management. Aerial or high-resolution satellite imagery give fast access to regional terrain and flood information, often in combination with additional GIS data such as terrain models and water resources. As an example, heavy floods have repeatedly affected the German river Elbe region. Precise and up-to-date maps are therefore indispensable for the prediction of water levels and streams, but also for disaster management and rebuilding. Figure 3 shows an aerial image of the Elbe river from August 2003 taken by a digital aerial camera ZI DMC.



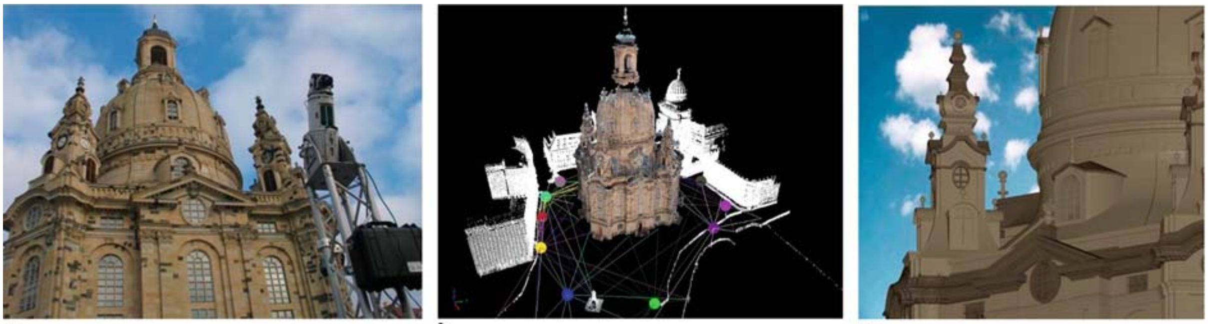
Photogrammetric Applications, Figure 3 Original image (**a**) and reconstructed groyne (**b**) (BfG Koblenz, EFTAS Münster)

Photogrammetric Applications, Figure 4 Original images (**a**–**i**) and resulting orthophoto mosaic (**j**)(IAPG Oldenburg)

These images have been used to measure digital terrain models of the river banks in order to complement airborne laser-scanning data which was acquired during high water. As an example, Fig. 3 right shows a 3-D model of a groyne as part of a larger terrain model that has been reconstructed from imagery.

Close-Range Applications

In contrast to aerial photogrammetry, the application areas in close-range photogrammetry are much broader. The most common applications address architecture and cultural heritage, industrial production control and quality assur-



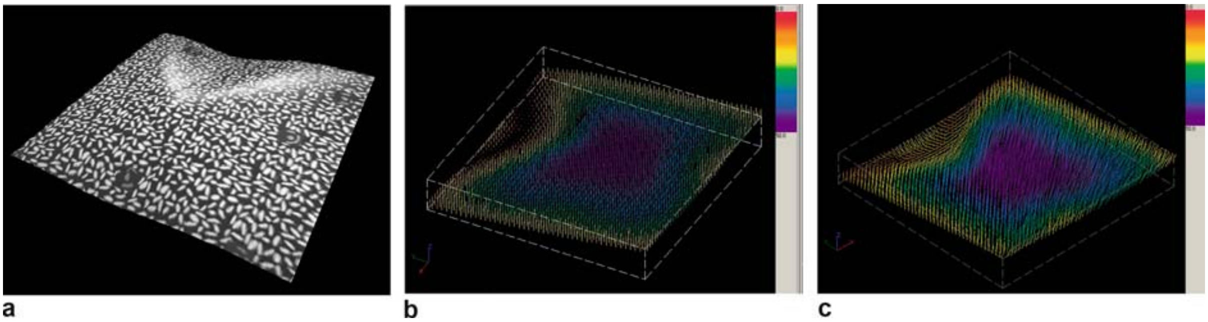
a b c

Photogrammetric Applications, Figure 5 Original image (**a**), 3-D point cloud (**b**) and example visualization (**c**) (Riegl)

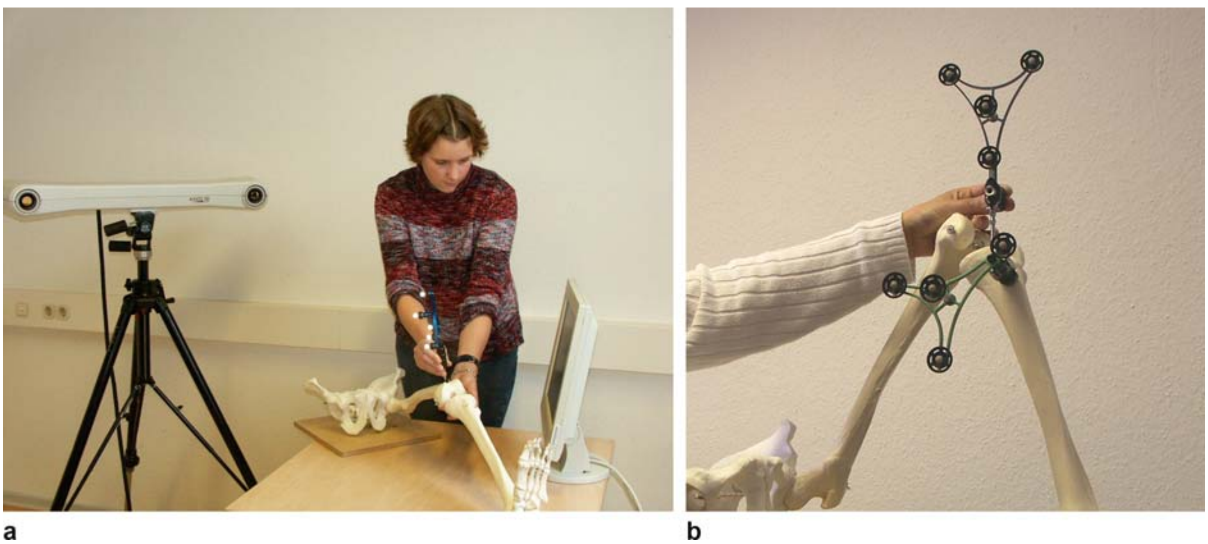
Photogrammetric Applications, Figure 6 Accident scene recording and photogrammetric processing (Photometrix)

ance, medicine, forensic and scientific applications. Imaging systems range from simple video to high-resolution digital cameras, from panoramic to high-speed cameras. Cameras are configured as single cameras or as multicamera setups. Imaging systems for the close range are often not designed as highly stable metric cameras, and hence they often must be calibrated either at shorter time intervals, or simultaneously with bundle adjustment for object reconstruction. The hybrid combination of cameras with other sensors, e. g., terrestrial laser scanners, is of increasing interest.

Close-range systems can be categorized into online and offline systems. Online systems generate 3-D data within a continuous data flow directly on the object site, e. g., for



Photogrammetric Applications, Figure 7 **a**–**c** High-speed image sequence processing for dynamic deformation analysis



Photogrammetric Applications, Figure 8 Stereo navigation system (**a**) and hand-held probe (**b**)

the navigation of tools in medical applications, or for the control of machines and robots in industry. In offline systems, image acquisition is often separated from image processing and object reconstruction. Hence, both parts can be performed in different locations, at different times and by different people. Products of close-range photogrammetry range from simple 3-D coordinates to process parameters, from free-form surface models to animated 3-D objects in static or dynamic environments.

The following examples cover four major application areas of close range photogrammetry, namely architecture, forensic analysis, industry, and medicine. Additional examples are given in [1,2,3,4,5,6,7].

Visualization of Architectural Objects Recording and visualization of buildings, archaeological sites or cultural heritage objects is one of the traditional photogrammetric applications. Besides 2-D drawings and plans, an increasing demand on rectified orthoimagery and 3-D models can be observed. Figure 4 shows a high-resolution image mosaic that consists of nine digital images with 4,000 × 4,000 pixels each.

Through a combination of close-range photogrammetry and terrestrial 3-D laser scanning it is possible to measure the complex surface shapes that exist in diverse forms for buildings, industrial process plants, archaeological excavations and sculptures. The example in Fig. 5 shows the Frauenkirche in Dresden, recorded by digital images, as well as airborne and terrestrial laser scans from multiple survey stations. The end result is a realistic, textured 3-D model of the church created by 3-D monoplotting.

Accident Recording Recording of traffic accidents is often characterized by difficult imaging configurations such as weak intersections of image rays or complex object scenes. The desired use of consumer digital cameras requires powerful and robust image calibration and orientation. Figure 6 shows an example of an accident scene and a subsequent scene reconstruction processed by the iWitness (Photometrix) software package.

Dynamic Surface Reconstruction Figure 7 shows the results of a dynamic deformation analysis of a car body part. The scene has been recorded by two high-speed cameras with a frame rate of 1,000 Hz. The object surface has been prepared by an artificial pattern in order to provide sufficient image texture for matching. In each epoch the surface is reconstructed by stereocorrelation following physical surface points through the image sequence.

Medical 3-D Navigation Figure 8 displays a stereo-camera system based on two video cameras (AXIOS 3D Services) that is used in medical applications for the measurement of the body and navigation of tools for computer assisted surgery. Usually this kind of system guides the surgeon for precise handling of surgical tools with respect to other tools, or part of the human body. The typical accuracy ranges from 3 mm down to 0.3 mm in a 1 m^3 measurement volume.

Future Directions

Photogrammetry serves as a flexible measurement tool in many different application fields. For both major areas, namely geotechnology and close-range applications, significant market growing rates of 15% per year and more have been predicted. It is therefore obvious that photogrammetry and 3-D image processing are fundamental upcoming technologies for a broad variety of applications.

Cross References

► Photogrammetric Products
► Photogrammetric Sensors
► Visualizing Constraint Data

Recommended Reading

1. Atkinson, K.B. (ed.): Close Range Photogrammetry And Machine Vision. Whittles, Caithness, UK (1996)
2. Fryer, J.G., Mitchell, H.L., Chandler, J.H.: Applications of 3D Measurement From Images. Whittles, Caithness, UK (2006)
3. Karara, H.M. (ed.): Non-Topographic Photogrammetry. 2nd edn. American Society for Photogrammetry and Remote Sensing. Falls Church, Virginia (1989)
4. Luhmann, T. (ed.): Nahbereichsphotogrammetrie in der Praxis. Wichmann, Heidelberg (2002)
5. Luhmann, T., Robson, S., Kyle, S., Harley, I.: Close-Range Photogrammetry. Whittles, Caithness, UK (2006)
6. Madani, M.: Photogrammetric applications. In: McGlone (ed.) Manual of Photogrammetry, 5th edn., pp. 1015–1104. American Society for Photogrammetry and Remote Sensing, Bethesda, Maryland (2004)>
7. ISPRS Commission V Close-Range Sensing: Analysis and Applications. http://www.commission5.isprs.org/. Accessed 20 Aug 2007

Photogrammetric Cameras

► Photogrammetric Sensors

Photogrammetric Images

► Photogrammetric Sensors

Photogrammetric Methods[1]

OLAF HELLWICH
Computer Vision and Remote Sensing,
Berlin University of Technology, Berlin, Germany

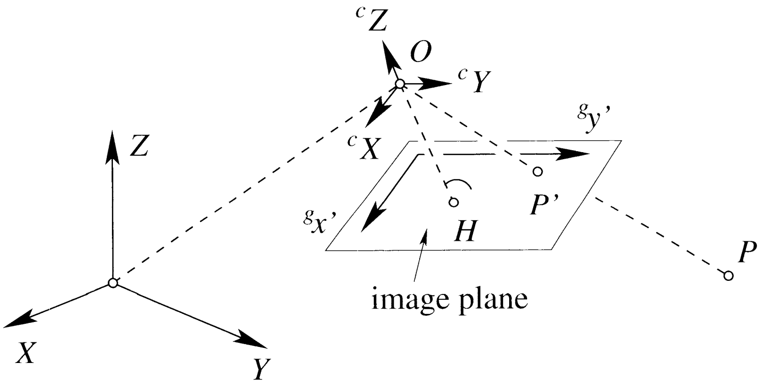
Synonyms

Camera model; Sensor orientation; Object reconstruction; Methods of photogrammetry; Single image; Image pair; Image triplet; Multiple-image bundle block; Bundle adjustment; Central projection; Central perspective; Fundamental matrix

Definition

The classical task of photogrammetry is the recovery of information from images of a scene [1]. This entry mathematically describes the geometry of single-perspective images, image pairs, image triplets, and blocks of several images. The notion “image” is very broad. Most widely used are images of frame cameras, i. e., the image is a 2D mapping from 3D object to 2D image space where the entire frame is exposed simultaneously through a lens. It is assumed that a unique projection center exists, so that the light rays between object and image points pass through a single point. In some cases, in particular when generating 2D images with a sweeping line camera, a unique projection center for the whole image does not exist. However, in all cases one assumes that one is able to determine a projection ray for a measurable image point in order to infer 3D information in object space from 2D measurements.

[1] This entry summarizes contents from the Manual of Photogrammetry [5]



Photogrammetric Methods, Figure 1 Mapping with a digital camera, object coordinate system $[X, Y, Z]$, projection center O, camera coordinate system $[{}^cX, {}^cY, {}^cZ]$, image or sensor coordinate system $[{}^gx', {}^gy']$, principal point H, object point P, image point P' [5]

Historical Background

In 1492 Leonardo da Vinci graphically demonstrated optical projection. Albrecht Dürer constructed mechanical devices to do perspective drawings of natural and studio scenes. In his classical treatise *The Free Perspective*, Henry Lambert dealt with the concept of inverse central perspective and space resection of conjugate image rays. It contained the geometric fundamentals of the process that 100 years later was named photogrammetry. With prophetic insight Guido Schreiber had rendered a treatise in 1829 on *The Process and Formulae for Air Topographic Equations and Determination of the Camera Station*, envisioning the time when Earth's image would be produced from a bird's-eye view. In 1849 Aimé Laussedat, an officer in the Engineering Corps of the French Army, embarked upon a determined effort to prove that photography could be used with advantage in the preparation of topographic maps. His work in this field was so complete that the principles demonstrated by practical applications are still in use. Not much later Ernst Abbe, cofounder of the Zeiss Works, placed the design of optical elements and their combination on a rigorous mathematical basis. In 1893, Albrecht Meydenbauer published a paper on the new method of photographic surveying in which the first use of the word photogrammetry appears. By the end of the 1930s, the semicomputational processes of the early days had fully been replaced by the optomechanical process of orienting stereo imagery to form a stereo model. This situation should change again with the advent of computers. After World War II, Hellmut Schmid developed the principles of multistation analytical photogrammetry. He rigorously applied the least-squares method to the simultaneous orientation of any number of photographs with a complete study of error propagation. In the 1960s there was considerable activity in developing and implementing practical adjustment algorithms for aerotriangulation, e. g., Duane Brown came up with an elegant and general treatment of least-squares adjustment and error propagation leading to computer programs, e. g., for extraterrestrial missions like Apollo. In the 1990s concepts of algebraic projective geometry were used to derive general direct solutions for photogrammetric problems advantageous in the automation of image analysis using uncalibrated low-cost cameras.

Scientific Fundamentals

In the following, a geometric model of the projection of points into the image generated by a real camera is formulated. It allows the projection process to be inverted to infer the spatial direction to 3D points from their observed images, and to use this to determine the spatial position of the camera and the 3D position of the observed points [2,3].

For modeling the projection, points are represented in three coordinate systems (Fig. 1): the object coordinate system S_o with object coordinates $\boldsymbol{x} = (X, Y, Z)^t$, the camera coordinate system S_c with camera coordinates ${}^c\boldsymbol{x} = ({}^cX, {}^cY, {}^cZ)^t$, and the sensor coordinate system S_g with image coordinates ${}^g\boldsymbol{x}' = ({}^gx', {}^gy')^t$. It is assumed that all coordinate systems are Euclidean and right handed.

The exterior orientation transforms the coordinates $\boldsymbol{x}_P$ of a point P from the object coordinate system S_o into the camera system S_c. This can be achieved in two steps by a translation of the object coordinate system S_o into the projection center O, and a rotation of the coordinate system S_o into the system S_c. The rotation matrix $\boldsymbol{R}$ can be represented by three independent parameters. In Euclidean coordinates:

$$ {}^c\boldsymbol{x}_P = \boldsymbol{R}(\boldsymbol{x}_P - \boldsymbol{x}_O)\,. \quad (1)$$

Often, camera models are formulated using homogeneous coordinates. Homogeneous coordinates $\mathbf{e}$ of an entity are invariant with respect to multiplication by a scalar $\lambda \neq 0$, thus that $\mathbf{e}$ and $\lambda\,\mathbf{e}$ represent the same entity. For instance, a 3D point with Euclidean coordinates $\boldsymbol{x} = (X, Y, Z)^t$ has homogeneous coordinates $\mathbf{x} = (U, V, W, T)^t$ which are related by:

$$\mathbf{x} = \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix} = \begin{bmatrix} U \\ V \\ W \\ T \end{bmatrix} = \begin{bmatrix} XT \\ YT \\ ZT \\ T \end{bmatrix}.$$

In homogeneous coordinates (1) reads:

$$\begin{aligned} {}^c\mathbf{x}_P &= \begin{bmatrix} {}^c\boldsymbol{x}_P \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \mathbf{0} \\ \mathbf{0}^t & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{x}_0 \\ \mathbf{0}^t & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_P \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{R} & -\boldsymbol{R}\boldsymbol{x}_0 \\ \mathbf{0}^t & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_P \\ 1 \end{bmatrix} = {}^c\mathbf{M}\mathbf{x}_P\,. \end{aligned} \tag{2}$$

When mapping with an ideal central perspective camera having a distortion-free lens and a planar sensor area, the Euclidean sensor coordinate system S_{g} is centered at the point on the image plane closest to the projection center, i. e., at the principal point, and the axes of this system are parallel to the axes of the camera coordinate system S_{c}. Then the homogeneous coordinates of an image point are:

$$\begin{aligned} {}^c\mathbf{x}'_{\overline{P}} &= \begin{bmatrix} {}^cu'_{\overline{P}} \\ {}^cv'_{\overline{P}} \\ {}^ct'_{\overline{P}} \end{bmatrix} \\ &= \begin{bmatrix} c & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^cX_P \\ {}^cY_P \\ {}^cZ_P \\ 1 \end{bmatrix} = {}^c\mathbf{P}_c\,{}^c\mathbf{x}_P\,. \end{aligned}$$

The 3×4 matrix ${}^{\mathrm{c}}\mathbf{P}_{\mathrm{c}}$ performs the projection from the object point P given in the camera coordinate system into the point $\overline{P\prime}$ in an ideal sensor coordinate system. For the ideal camera ${}^{\mathrm{c}}\mathbf{P}_{\mathrm{c}}$ contains only one parameter defining its interior orientation, namely its principal distance c.

Using (1) and (2), the composed mapping from object space to image space with an ideal camera is expressed as:

$$\begin{aligned} {}^c\mathbf{x}' &= {}^c\mathbf{P}\mathbf{x} = {}^c\mathbf{P}_c\,{}^c\mathbf{M}\mathbf{x} \\ &= \begin{bmatrix} c & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & -\boldsymbol{R}\boldsymbol{x}_O \\ \mathbf{0}^t & 1 \end{bmatrix} \mathbf{x}\,. \end{aligned}$$

Introducing a calibration matrix:

$${}^c\mathbf{K} = \begin{bmatrix} c & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

the projection reads

$${}^c\mathbf{x}' = {}^c\mathbf{K}\boldsymbol{R}\,[\mathbf{I}\,|-\boldsymbol{x}_O\,]\,\mathbf{x}$$

The Euclidean coordinates of the image point are given by the so-called collinearity equations:

$${}^cx' = c\frac{r_{11}(X - X_O) + r_{12}(Y - Y_O) + r_{13}(Z - Z_O)}{r_{31}(X - X_O) + r_{32}(Y - Y_O) + r_{33}(Z - Z_O)}$$

$${}^cy' = c\frac{r_{21}(X - X_O) + r_{22}(Y - Y_O) + r_{23}(Z - Z_O)}{r_{31}(X - X_O) + r_{32}(Y - Y_O) + r_{33}(Z - Z_O)}$$

derived by dividing through the third component.

In order to model a real camera the projection model is extended in two steps:

- All terms which guarantee the projection to be straight-line-preserving are added.
- Then additional terms allowing the modeling of general cameras still having a unique projection center are introduced [4].

Note that the ideal point $\overline{P'}$ is assumed to be identical to the measurable point P' observed in a skew coordinate system being related to the ideal coordinate system by an affine transformation. The parameters of this transformation are the translation of the coordinate system into the principal point $\left(x'_H, y'_H\right)^t$ of the sensor coordinate system S_{g}, the correction of the scale of the y' coordinates with respect to the x' coordinates by the factor $1 + m$, and the shear of the ${}^{\mathrm{c}}y'$ axis $s = \tan(\alpha)$, where α is the shear angle. Including this transformation into the calibration matrix results in:

$$\mathbf{K} = \mathbf{H}_c\,{}^c\mathbf{K} = \begin{bmatrix} c & cs & x'_H \\ 0 & c(1+m) & y'_H \\ 0 & 0 & 1 \end{bmatrix}.$$

The final projection then reads as:

$$\mathbf{x}' = \mathbf{P}\mathbf{x} \tag{3}$$

with the homogeneous projection matrix:

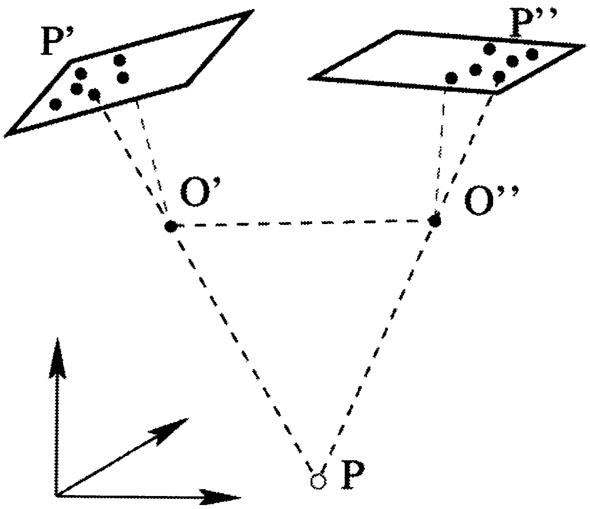
$$\mathbf{P} = \mathbf{K}\boldsymbol{R}\,[\mathbf{I}\,|-\boldsymbol{x}_O\,]$$

which contains 11 parameters, namely the 5 parameters of the interior orientation in matrix $\mathbf{K}$ and the 6 parameters of the exterior orientation.

The mapping Eq. 3 with the elements p_{ij} of P is explicitly given as:

$$\begin{aligned} x' &= \frac{p_{11}X + p_{12}Y + p_{13}Z + p_{14}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}} \\ y' &= \frac{p_{21}X + p_{22}Y + p_{23}Z + p_{24}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}}\,. \end{aligned}$$

It is called the direct linear transformation (DLT) [5], as it directly relates the Euclidean coordinates of the object points and measurable sensor coordinates of the image points of a straight-line-preserving camera.



Photogrammetric Methods, Figure 2 Geometry of an image pair [1]

In the above-mentioned second step an additional homography-like transformation depending on the local position ${}^{c}\boldsymbol{x}'$ in the image and additional parameters $\boldsymbol{q}$ is introduced:

$$ {}^{g}\mathbf{x}' = {}^{g}\mathbf{H}({}^{c}\boldsymbol{x}')\mathbf{x}' $$

resulting in general image coordinates ${}^{g}\mathbf{x}'$ and using

$$ {}^{g}\mathbf{H}({}^{c}\boldsymbol{x}') = \begin{bmatrix} 1 & 0 & \Delta x'({}^{c}\boldsymbol{x}', \boldsymbol{q}) \\ 0 & 1 & \Delta y'({}^{c}\boldsymbol{x}', \boldsymbol{q}) \\ 0 & 0 & 1 \end{bmatrix} $$

where the terms $\Delta\mathbf{x}'({}^{c}\boldsymbol{x}', \boldsymbol{q})$ and $\Delta y'({}^{c}\boldsymbol{x}', \boldsymbol{q})$ are local corrections depending on $\boldsymbol{q}$ usually defining polynomials. Then real cameras showing distortions that do not preserve straight lines, most notably radial distortions of the lens system, can also be modeled.

The three dimensional object structure can be inferred from two images taken from two different places. For this purpose corresponding points P'_i and P''_i in the two images are measured. For a perfect orientation of the cameras the two corresponding rays $P'O'$ and $P''O''$ from the image points through the projection centers would intersect in the object point P. This is the so-called coplanarity constraint, since the corresponding rays of an oriented image pair are coplanar (Fig. 2).

An explicit expression for the coplanarity constraint for the relative orientation of two cameras is given by

$$ \mathbf{x}'^{t}\,\mathbf{F}\mathbf{x}'' = 0\,. \tag{4} $$

For a derivation from the projections of the two cameras according to Eq. 3 see [1]. The 3×3 fundamental matrix $\mathbf{F}$ is determined by seven independent parameters, as it is homogeneous and singular. Therefore, only seven corresponding points are necessary to determine its elements.

The point P'' in the second image corresponding with a point P' in the first image is located on a straight line. This line, called the epipolar line, is very helpful when searching for corresponding so-called homologous points. The underlying geometry is the epipolar geometry (Fig. 3). The epipolar plane $\varepsilon\,(P)$ defined by the projection centers O' and O'' and the object point P intersects the image planes ε' and ε'' at the epipolar lines $l'(P)$ and $l''(P)$. The epipolar lines of all object points intersect at the epipoles E' and E''. These entities can be determined using the projection matrices or the fundamental matrix. Most importantly, due to the coplanarity constraint and the incidence of image points and epipolar lines $\mathbf{x}'^{t}\,\mathbf{l}' = 0$ and $\mathbf{x}''^{t}\,\mathbf{l}'' = 0$, the epipolar lines are given by

$$ \mathbf{l}' = \mathbf{F}\mathbf{x}'' \quad \mathbf{l}'' = \mathbf{F}^{t}\mathbf{x}'\,. $$

The relative orientation of three images gives constraints on all image coordinates involved. As the previous treatment of epipolar geometry shows, this is not the case for the image pair which only gives constraints in one direction. Therefore, it is useful to investigate the geometry of the image triplet expressed by the so-called trifocal tensor. It can be used to predict points and lines given in two images in the third one. The prediction of a line l' in the first image from given lines l'' and l''' in the other images can be obtained from

$$ \mathbf{l}' = \begin{bmatrix} \mathbf{l}''^{t}\mathbf{T}_1\mathbf{l}''' \\ \mathbf{l}''^{t}\mathbf{T}_2\mathbf{l}''' \\ \mathbf{l}''^{t}\mathbf{T}_3\mathbf{l}''' \end{bmatrix} \tag{5} $$

where $\mathbf{T}_{\mathrm{i}}$ are the three 3×3 trifocal matrices stacked in the trifocal tensor $\mathbf{T}$. In simplified notation, (5) is expressed as [6,7]

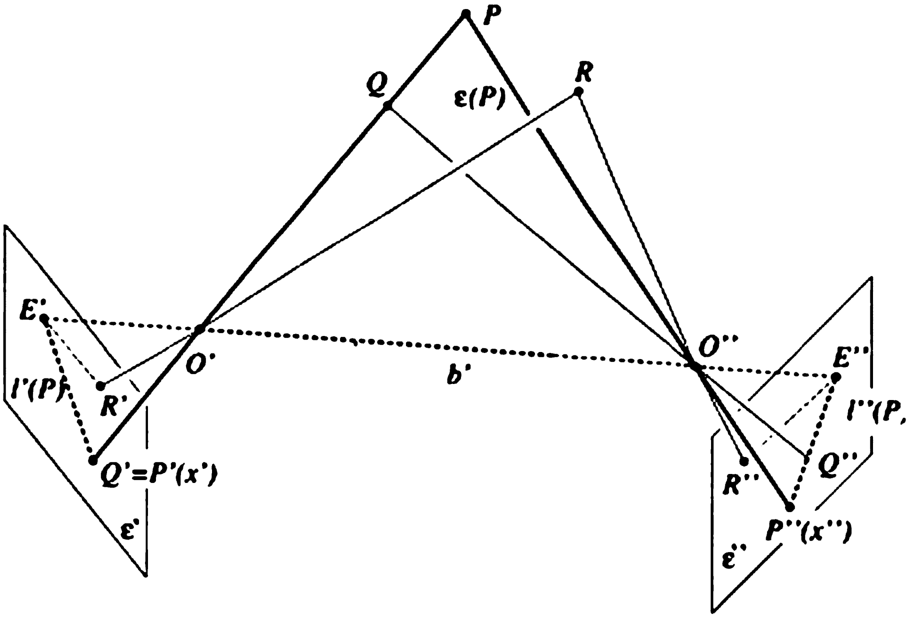
$$ \mathbf{l}' = \mathbf{T}(\mathbf{l}'', \mathbf{l}''')\,. $$

Similarly, points can be predicted.

There exist relations between projection matrices, fundamental matrices and trifocal tensors. Owing to spatial limitations these derivations are not given here.

The three-dimensional position of object points can be determined by intersecting the rays defined by their image points and the corresponding projection centers. This is called photogrammetric triangulation and frequently based on multiple images. As the imaging rays of a camera form a bundle and the images overlappingly cover the object

P



Photogrammetric Methods, Figure 3 Elements of the epipolar geometry: epipolar plane $\varepsilon(P)$ through $O'O''P$, with the epipoles E' and E'' as images of the other projection center (O' or O''), the epipolar lines $l'(P)$ and $l''(P)$ which are the intersections of the epipolar plane $\varepsilon(P)$ and the image planes ε' and ε''. The epipolar planes build a pencil of planes with the base line $b' = O'O''$ as axis, e. g., induced by a different point R. Therefore the epipolar lines also build a pencil of lines with the epipoles as carrier. Observe, P' does not allow inference of where P sits on the projecting line. Point Q, also mapping to P', however, has a different image Q'' inducing the epipolar line $l''(P) = (E''Q''P'')$ in the other image [1]

space, this process is also called bundle block triangulation. It is usually formulated as a least-squares adjustment problem often termed bundle adjustment with, e. g., image coordinates and object space coordinates of control points as observations, and orientation parameters of the cameras and object space coordinates as unknowns. For instance, the collinearity equations, (3), could serve as observation equations in a Gauss–Markov adjustment model.

Key Applications

Photogrammetric 3D reconstructions are applied in various application such as acquisition of geoinformation, topographic mapping and terrain model generation using airborne imagery, and all kinds of close-range 3D reconstructions, e. g., in architecture, archeology, and engineering, as well as many applications in computer vision.

Future Directions

Recent developments include mathematical modeling and application of nonpinhole and uncalibrated cameras in real-time environments requiring direct (explicit) solutions for unknown parameters.

Cross References

► Data Acquisition, Automation
► Laser Scanning
► Photogrammetric Applications
► Photogrammetric Products

Recommended Reading

1. McGlone, J.C. (ed.): Manual of Photogrammety, 5th edn. American Society for Photogrammetry and Remote Sensing. Bethesda, Maryland (2004)
2. Ma, Y., Soatto, S., Kosecka, J., Sastry, S.S.: An Invitation to 3-D Vision. Springer Verlag, New York (2004)
3. Mikhail, E.M., Bethel, J.S., McGlone, J.C.: Introduction to Modern Photogrammetry. Wiley, New York (2001)
4. Swaminathan, R., Grossberg, M.D., Nayar, S.K.: A perspective on distortions. Proc. Comput. Vision Pattern Recog. 2:594–601 (2003)
5. Abdel-Aziz, Y.I., Karara, H.M.: Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. Proc Symp. Close-range Photogrammetry, pp. 1–18, Am. Soc. Photogrammetry (1971)
6. Faugeras, O.D., Luong, Q.T., Papadopoulo, T.: The Geometry of Multiple Images. MIT Press, Cambridge, MA (2001)
7. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, New York (2003)

Photogrammetric Products

J. Chris McGlone
SAIC, Chantilly, VA, USA

Synonyms

Map data; Cartographic data; Ortho-image; DEM; Digital elevation model; Elevation reference surface (datum); Digital surface model; TIN; Triangulated irregular network; Georectified; Ortho-mosaic; Oblique images; Root-mean-square error; RMS error; National map accuracy standard; 3D models; Photo-textured

Definition

A photogrammetric product is a representation of aspects of a scene derived from imagery of the scene. The representation may be geometric and include point coordinates, object geometry or measurements, or other attributes derivable from image geometry. In some cases, qualitative object properties may be added onto the basic geometric data.

Historical Background

Traditionally, photogrammetric products meant hardcopy maps depicting elevation as contours and features as lines. With the advent of digital softcopy photogrammetry for production and the widespread adoption of GIS to utilize cartographic data, emphasis has shifted almost exclusively to products in digital form. The increasing availability of digital imaging sensors has accelerated this trend. Indeed, the most rapidly growing types of photogrammetric products involve digital imagery, geo-located and processed for various GIS and consumer applications and delivered over the Internet.

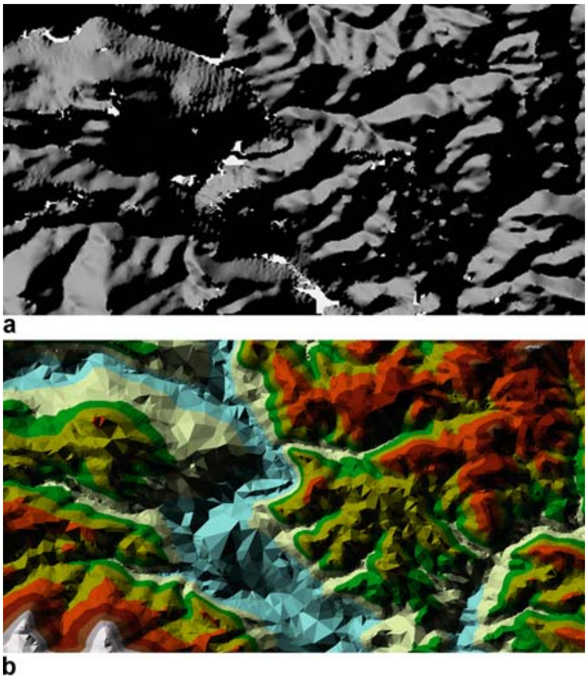
Scientific Fundamentals

The basis of a GIS is its geospatial information; photogrammetry is unique in its ability to provide accurate spatial data with high information content over wide areas. The orthoimage is a prime example; when used as a GIS base map, it provides a foundation for positional information as well as context for the display and interpretation of other data.

Elevation Products

Elevation products [1] represent the elevation of the earth's surface. Both raster and triangulated irregular network (TIN) representations are used, with the choice depending on the particular application.

Raster representations [1,2] are the most common, since they can be displayed and manipulated using standard image processing software and hardware. DEMs can be defined relative to any coordinate system, either projected coordinate systems such as UTM or directly in latitude-longitude. DEMs are described by their resolution or *post spacing*, the distance between adjacent elevation samples. For instance, USGS DEMs are usually described as 30-meter or 10-meter DEMs (for those in UTM), while NGA Digital Terrain Elevation Data (DTED) comes with 3-arc-second or 1-arc-second (latitude-longitude) post spacings. European DEM products include, from the UK, Land-Form PROFILE® Plus (2 m grid, 0.5 m RMSE for urban and flood plain areas up to 10 m grid with 2.5 m RMSE for mountain and moorland areas), from Germany, ATKIS DGM5 or DGM 25 with 5 m (not available everywhere) or 25 m grid spacing, and from France, BD ALTI® with 50 m grid spacing.



Photogrammetric Products, Figure 1 **a** Elevation raster, shown in hill-shaded form. **b** TIN version of the raster

Another important property of DEM products is the elevation reference surface, or datum. DEMs in the past were referenced to local height datums relative to sea level, but today are usually referenced either to a global geoid model (e. g., GEOID99) or to a reference ellipsoid (e. g., WGS84).

TIN representations [1,3] consist of a set of irregularly-distributed points connected by edges to form a surface consisting of connected triangles. TINs are typically more efficient than rasters in terms of the storage space required for an equivalent level of detail or accuracy, since more points can be concentrated in complex areas and fewer points used in flat areas, although three coordinates must be stored for a TIN point versus only the Z coordinate for raster representations. TIN points can be placed at the edges of breaks or in the bottoms of depressions in the terrain, whereas a raster's fixed sampling interval may not capture such terrain detail. In some cases rasters are augmented by breaklines which depict abrupt changes in surface slope.

TINs are more complicated to display than rasters since 3D graphics are required instead of simple raster displays.

P

TINs work well for graphics and simulation applications since current graphics cards are highly optimized to deal with sets of triangles. Exploitation is also more complicated, since determining the elevation at a given X,Y coordinate requires first identifying the triangular face of the TIN containing it, then interpolating the elevation from the three vertices of the face.

Contours (lines of equal elevation) [1,3,4] were formerly the standard elevation representation, due to ease of visual interpretation and their suitability for photogrammetric extraction. However, they are now secondary products derived from rasters or TINs as required.

Most elevation models depict the earth's surface as it would appear without buildings or vegetation and are referred to as digital elevation models (**DEM**) or digital terrain models (**DTM**). Constructing a DEM requires manual interpretation of the scene to remove non-terrain objects and to estimate the terrain elevation. Current automated processes such as automated stereo correlation or 3D sensors such as LiDAR or IFSAR represent the first (reflective) surface, containing buildings and the tops of trees or vegetation. These are referred to as digital surface models (**DSM**) and may be used for orthoimage production. Automated editing methods are somewhat successful in reducing DSMs to DEMs, but some manual editing is still required.

The majority of DEMs are currently produced by photogrammetric methods, although 3D sensors such as LiDAR are being rapidly adopted. To produce a DEM, the operator views the scene in stereo and places a 3D measuring dot superimposed on the model on the ground at the desired post spacing. Alternatively, the operator may capture 3D points at representative locations on the terrain and thereby generate a TIN from which a DEM can be interpolated if required. Automated stereo methods replace the operator by performing the stereo matching using image correlation techniques.

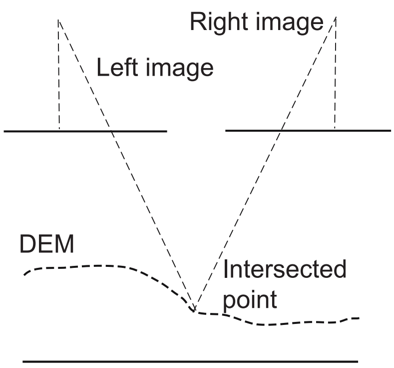
DEMs may be distributed in image formats, such as geotiff, or in special data formats such as USGS DEM or NGA DTED. DEM specifications typically specify the Z RMS error against some number of independently measured elevation points. Common DEM errors include noise spikes or pits due to measurement or processing errors. There may also be systematic offsets due to operator biases or caused by automated processes measuring the tops of vegetation instead of the ground surface.

Image Products Before an image can be used in a GIS, there must be some means to relate the locations of objects within the image to their locations in the world. This geometric relationship between a pixel in the image and a point on the ground is embodied in the *sensor model*, sometimes referred to as the image metadata. The sensor model includes:

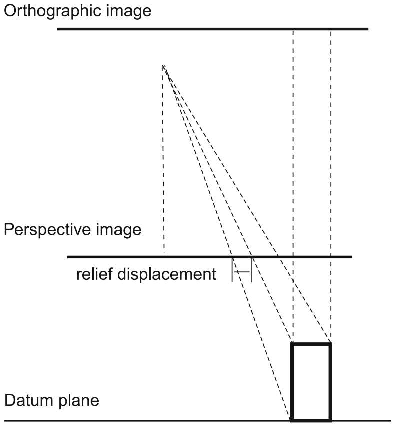
- the location of the perspective center of the sensor in world coordinates and the angular orientation of the camera with respect to the world coordinate system. This is sometimes referred to as the *exterior orientation*, since it places the sensor within an exterior reference system. For non-frame sensors, these parameters may be expressed as functions of time to model the path of the aircraft or the orbit of the satellite carrying the sensor.
- the *interior orientation*, the geometric parameters of the sensor itself. This includes the principal distance (the focal length for images at infinity) and the specification of origin, orientation, and scale of the image coordinate system.
- a set of equations, based on the principles of perspective geometry and using the parameters of exterior and interior orientation, which relates a point in the image to a point in the world.

Given the sensor model, one can model the path of a ray of light from a point in the world through the perspective center of the sensor and onto the imaging plane to calculate its image coordinates; alternatively, one can use the sensor model and the image coordinates to calculate the ray in space passing through the object in the world. Note that a single image can specify only the direction in space to an object: to calculate the 3D position of an object we must intersect rays from two or more sensors or else have external knowledge of the scene geometry, such as a digital elevation model, and intersect the ray with that surface to determine a 3D position.

Very few GIS include the capability to deal with the variety of sensor model types currently in use. Additionally, perspective effects present in unprocessed images make their combination with other types of data problematic. There-



Photogrammetric Products, Figure 2 Determination of the 3D position of a point by the intersection of rays from two images or by the intersection of a ray from either image with a DEM



Photogrammetric Products, Figure 3 Orthographic and perspective projections, showing relief displacement due to the height of the object

fore, images are usually processed to transform them into a more easily exploited form, both in terms of appearance and sensor model.

Rectified or **geo-rectified** imagery [2] is produced by reprojecting the image to a reference surface. Rectified frame photos were widely used in the past, since the film could be transformed into an equivalent vertical photograph using an analog rectifier. Digitally geo-rectified images are reprojected to a reference plane or, in the case of satellite imagery, the ellipsoid surface. This removes perspective effects, but does not correct for displacements due to differing elevations across the scene.

Orthorectified images (**orthoimages**) [2,4] are produced by transforming the original image into an orthographic projection. In an orthographic projection the projection direction is perpendicular to the datum plane, as in a map (Fig. 3), whereas in a perspective image objects above the datum plane are displaced proportional to their height (relief displacement). Since objects in an orthographic projection are shown at their true map locations, orthoimages are often used as base layers in GIS databases. Common examples of orthoimage products include the U.S. Geological Survey's Digital Ortho Quads, the UK Ordnance Survey Mastermap® Imagery Layer (0.25 m), the German ATKIS DOP (0.1–0.4 m), and the French BD Ortho® (0.5 m). One of the most visible current applications of orthoimagery is as a base for systems such as Google Earth, Microsoft Virtual Earth, and NASA WorldWind.

Orthoimage production requires the 3D coordinates of each point in the image, usually obtained by intersecting the image rays with a DEM of the scene. For each X,Y pixel location in the final orthoimage, the elevation is determined from the DEM and the coordinates are projected into the perspective image. The intensity value at that point in the orthoimage is then set to that of the perspective image. If multiple input images are available, the orthoimage intensity value can be determined as a combination of the various input images.

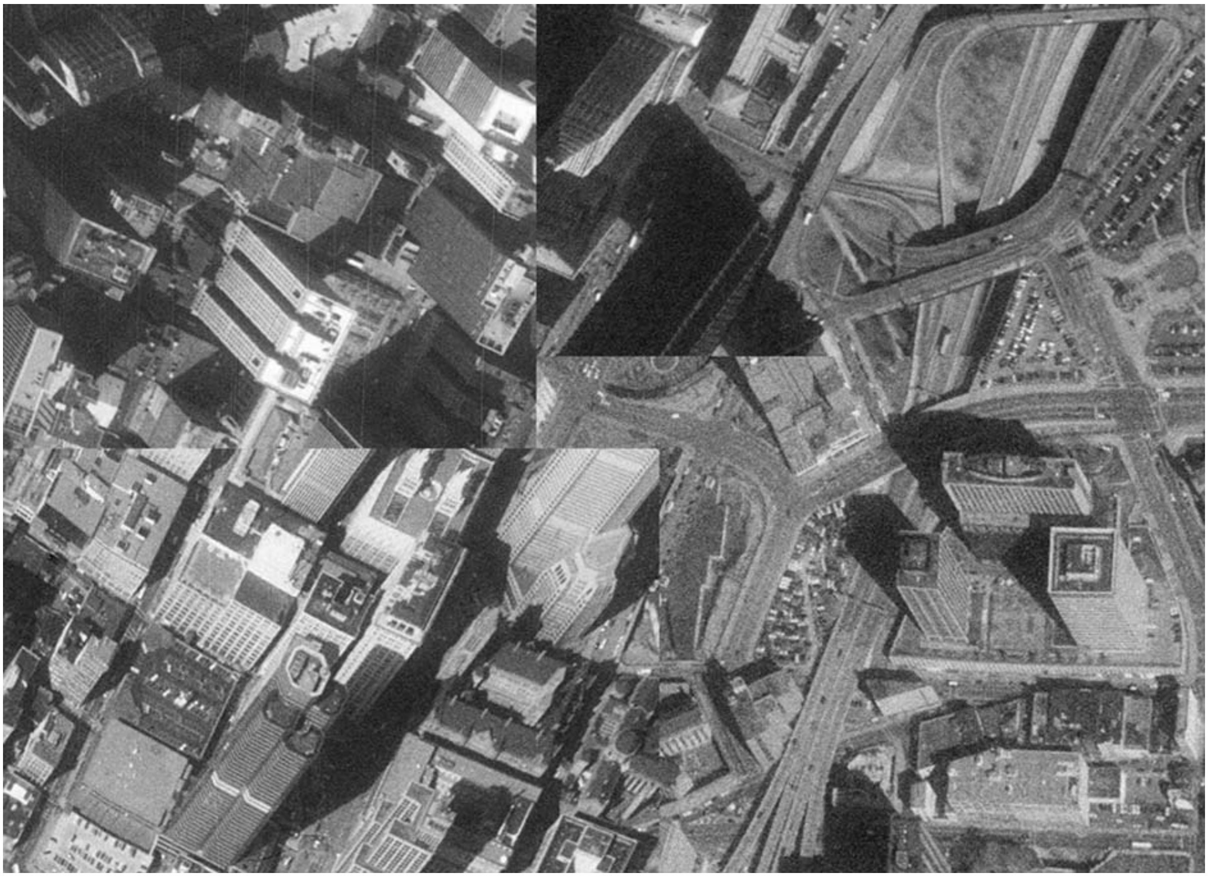
However, a digital elevation model contains the elevations of the terrain surface and not the elevations of structures on the terrain. Buildings in the scene will therefore appear to lean in the orthoimage (Fig. 4), due to the image perspective. Sometimes a digital surface model (DSM) is used, which contains the elevations of the terrain and objects on it. A DSM may be produced by photogrammetric methods or by direct 3D sensors such as LiDAR or IFSAR. Alternatively, 3D building models may be manually extracted and used in conjunction with the DEM to allow building roofs to be projected into their correct positions and occluded areas to be identified. Orthoimages produced in this manner are often referred to as **true orthoimages**. If multiple images are available, the area occluded by the building can be filled in from other viewpoints. To eliminate building shadows from the final image, shadows can be detected by comparing intensities among the images, or the shadow geometry may be predicted from the sun angle.

Most orthoimages are actually **orthomosaics** [2] produced from multiple images, permitting the coverage of large areas and the selection of the best image for any particular point. The images must be carefully blended for radiometry and color balance and the seam boundary between images must be carefully drawn to make it invisible.

Orthoimages are not suitable for all applications, since they show only building roofs and outlines which are hard to recognize from street level. Oblique aerial views (Fig. 5) show building facades and make building recognition and the determination of characteristics such as the number of floors much easier. Several companies now offer oblique aerial imagery covering sites from different angles, along with the associated sensor models and tools to enable their exploitation. The tools are designed to work either as plugins to GIS or to interoperate with GIS tools and allow measurements and positioning from the imagery. The positioning accuracy is typically limited by the precision of the navigation information.

Oriented image stereopairs may also be supplied, with their associated metadata, allowing exploitation within GIS using photogrammetric software designed to work within GIS packages. Several commercial satellite companies supply such stereopairs and support data, ready for exploitation by mapping companies. This is less common for aerial photography, since few users are equipped to do stereo extraction.

Distributing oriented imagery requires that the exploitation software implement the appropriate sensor model. Given the wide variety of sensor types currently in use, both aeri-



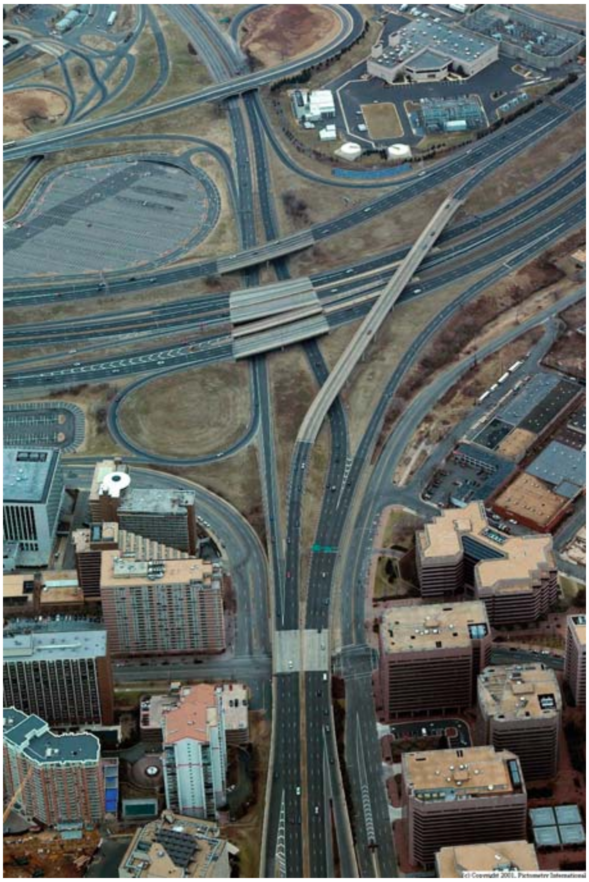
Photogrammetric Products, Figure 4 Apparent building lean in an orthomosaic, particularly evident at the mosaic seams. Notice that features at ground level are aligned

al and satellite, this poses a major implementation, verification, and maintenance burden. For this reason, there has been much work on "generic" or "replacement" sensor models, where the geometry of the sensor is modeled by a set of polynomials derived from the original physical model by the supplier of the imagery. Exploitation systems then only have to support the common polynomial model.

Image products may be produced in proprietary formats, but nearly all are available in standard formats. Orthoimages are widely distributed in geotiff format, which is based on the popular tiff image format and includes header tags which contain the coordinate and projection information. A newer standard is the JPEG2000 format, which uses the JPEG2000 image compression format along with header tags for geopositioning information. The US Dept. of Defense has defined the National Image Transmission Format (NITF), which allows for the inclusion of multiple images and their associated metadata as well as graphic overlay and text information within the same file structure.

Image product specifications have two main aspects, radiometric and geometric. Radiometric specifications are concerned with the appearance and interpretability of the image product. Good contrast and brightness are crucial; both qualities refer to the distribution of pixel values across the image histogram. Panchromatic (gray scale) images most often have 8 bits per pixel, meaning that they can represent 256 shades of gray. An image with good contrast will distribute the pixel values across nearly all 256 possible values, maximizing the visual information content. An image with good brightness level will have the values peaking near the middle of the histogram, instead of concentrating at one end or the other. Many digital sensors now collect data with 11 or 12 bits of radiometric resolution. This greater range must be mapped into the typical 8 bit display while preserving both fine detail and overall structure and contrast.

The color balance of the image is also important, both its relationship to the original colors in the scene (assuming that it is a true color image, as opposed to a false-color or



Photogrammetric Products, Figure 5 Oblique aerial image, supplied with exterior orientation parameters to allow measurement and positioning. Courtesy of Pictometry, Inc

infrared image) and its appearance on the computer monitor. Tasks requiring precise image interpretation based on color require the calibration of the monitor to color standards as well as correction for the color response of the sensor.

Improvements in digital sensors have greatly reduced the occurrence of image artifacts such as streaks or blooming due to overexposure, but there may still be issues related to the acquisition, such as haze, cloud cover, or bright spots resulting from the relative alignment of the sun and sensor.

The geometric accuracy of an image product is specified in terms of the error computed by comparing identifiable image points with independent coordinate measurements. These measurements will be in X and Y for orthoimage products, or in all three coordinates for oriented stereopairs. The accuracy is typically specified in terms of the root-mean-square (RMS) error over a given number of well-distributed check points. For instance, the USGS specification for Digital Ortho Quads requires that they meet National Map Accuracy Standards at 1:24,000 scale: 90 percent of the well-defined points tested must fall within 40 feet.

Vector Feature Products Vector products [3,4] may be considered "line drawings" of objects or areas in the scene. They are most often used to describe man-made objects such as roads or buildings, or to delineate areas. Vectors features have three aspects: geometry, topology, and attribution.

The geometry of vector features is usually expressed in terms of the coordinates of individual points, although sometimes the feature will reference a defined geometric form such as a circular arc, spline, or rectangle. Features may be points, lines, or polygons. Lines may be "polylines" consisting of multiple connected lines. A 2D feature has only X and Y coordinates; if the Z coordinate is included for each point the feature is often called "2.5D," since non-horizontal faces of 3D objects such as buildings are not explicitly represented. A true 3D representation contains information on object faces, such as the direction of the outward-facing normal vector (cf. below).

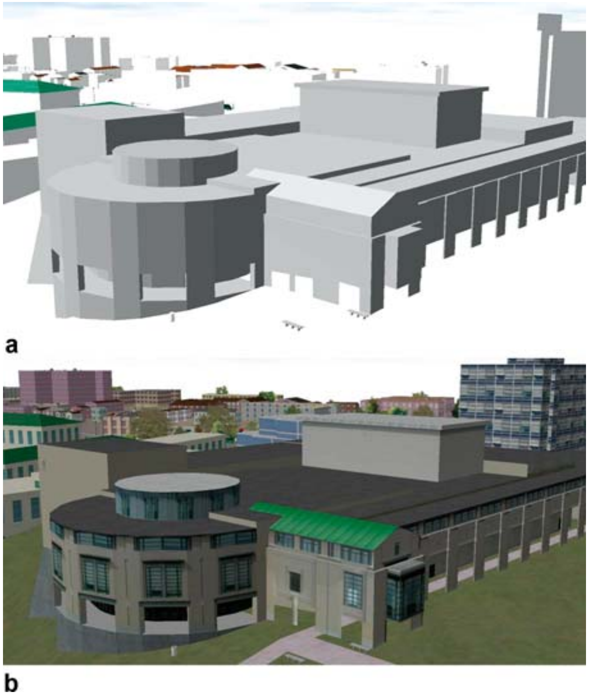
A topological representation allows reasoning over spatial relationships between objects, such as "adjacent to," "inside," or "connected to." For instance, for a given line, connected lines and adjacent faces can be easily determined. The elements of a topological representation are nodes, points, lines, and faces. Nodes are connections between lines, as opposed to points which only indicate position. Lines consist of ordered sets of points, and connect to other lines only at nodes. A line has a left and right side and may be the border of a face. Faces may contain other faces and also be contained within larger faces.

Topological encoding enables reasoning about the properties of a scene. For instance, a topologically-encoded road network allows reasoning about routes, through the connectivity information, or about the access of parcels of land represented as faces to the road network, since the neighboring face of each line is defined. The collection of topological information also allows collected data to be checked as it is added.

Vector features are usually collected with *attribution* which stores visible or inferred properties of the feature. Road features may contain attributes which indicate the type of surface (concrete, asphalt, gravel), the number of traffic lanes, or the type of road; buildings attributes may include an inferred use (residential, commercial, factory) or type of construction (brick, frame).

Specifications for vector data typically concern spatial accuracy, the types of features to be captured, and the degree of detail captured. Some examples of current vector datasets include USGS Digital Line Graph, UK OS

P



Photogrammetric Products, Figure 6 **a** 3D building model without phototexture. **b** Phototextured model

Mastermap® Topography Layer, German ATKIS Basis-DLM and DLM50, and French BD TOPO®.

3D Models An increasingly-common product is 3D models [3], used for both GIS and visualization applications. Most photogrammetric 3D models are collected as wireframes, which are well suited for buildings composed of planar surfaces, although some applications use combinations of 3D geometric primitives. If 2D building footprints are available, they may be extruded upward to the measured building height to produce a 3D model, but in most instances 3D models are extracted by measuring corner points.

The photogrammetric extraction of 3D models is extremely labor intensive. While some semi-automated systems are being used in production, the problem for general buildings is so complicated that efforts at fully-automated extraction have not yet been successful. The main cost driver is the level of detail which must be represented. For telecommunications applications concerned with signal propagation, fairly coarse models are acceptable, while planning or security applications often require extremely detailed models so that calculated lines of sight are accurate or building appearances are realistic.

Building models are often *phototextured,* (have imagery applied to the faces) for a more realistic appearance. The simplest texturing is done by applying repeating generic wall patterns to building faces. When more realism is required, actual imagery of the building is used. For purely visual applications, building phototextures can often be substituted for detailed geometry—i. e., instead of collecting detailed façade structure, an image of the façade applied to the model will give the impression of the geometry without the expense of detailed extraction. Aerial imagery seldom works well for texturing, since the viewing angle is nearly parallel to the building surfaces and results in a "smeared" appearance. Oblique aerial imagery or ground imagery is usually much more satisfactory.

3D formats for GIS use are problematic. Most common 3D formats are designed for visualization purposes, supporting geometry and textures but not attribution. The most common example is OpenFlight® by Multigen-Paradigm. Several vendors do offer proprietary solutions to allow attribute query on 3D representations.

Key Applications

Photogrammetric products provide the location basis for nearly all GIS applications, either as the primary source of the information or as the context and framework into which data is conflated and utilized.

Future Directions

The trend to 3D products, especially building models, can be expected to continue as more consumer applications based on such products are introduced. The visual aspects will become more important, especially for consumer applications where absolute cartographic accuracy is less important. Photogrammetric products are just one component of increasingly complex and multi-faceted databases, which may include addresses, demographic information, or commercial and advertising content.

There will be an ongoing competition between the automation of feature extraction processes and its outsourcing to countries with lower-cost labor. For many areas, the task will change from feature extraction to maintenance, update, and enhancement of existing GIS systems. Licensing of portions of large-area datasets maintained by one vendor, such as road or address databases, will become more common than the production of data for a single user over a specific area.

Product distribution will occur through a growing number of channels, especially outside the traditional GIS/mapping industry. Products will increasingly be embedded in consumer applications and targeted to non-cartographic users and applications, for instance, 3D building models delivered to cell phones for location-based services.

Cross References

► Photogrammetric Methods
► Visualizing Constraint Data

Recommended Reading

1. Maune, D. (ed.): Digital Elevation Model Technologies and Applications: The DEM Users Manual, 2nd edn. American Society for Photogrammetry and Remote Sensing, Bethesda (2006)
2. Miller, S.: Photogrammetric Products. In: McGlone, J.C. (ed.) Manual of Photogrammetry, 5th edn. American Society for Photogrammetry and Remote Sensing, Bethesda (2004)
3. Mikhail, E.M., Bethel, J., McGlone, J.C.: Introduction to Modern Photogrammetry, John Wiley and Sons, New York (2001)
4. Wolf, P.R., Dewitt, B.A.: Elements of Photogrammetry with Applications in GIS, McGraw-Hill, New York (2000)

Photogrammetric Sensors

MICHAEL CRAMER
Institute for Photogrammetry (ifp),
Stuttgart University, Stuttgart, Germany

Synonyms

Photogrammetric images; Photogrammetric cameras; Aerial imagery; Image acquisition; Air borne sensors; Remote sensing

Definition

Images are the main data source of photogrammetric data processing. Hence the sensors used for data acquisition are an elementary part of the photogrammetric processing chain. In general, images are taken by satellite, airborne, or terrestrial sensors for photogrammetric applications such as object and terrain modeling and acquisition of topographic data. This entry summarizes the state-of-the-art and focuses on the strong trend towards digital image recording.

Historical Background

The idea of using photographs for the reconstruction of the imaged objects was born almost at the same time as the invention of photography. Starting in the mid of the 19th century object coordinates were estimated from two dimensional images based on the fundamental equations of image geometry. The term "photogrammetry" appeared for the first time in 1867 [1]. In the beginning photogrammetric reconstructions were limited to terrestrial applications, although first experiments to obtain imagery from the air were already performed with balloons or kites long before 1900. The success of those attempts has been limited due to the restricted maneuverability of the camera carrier. The situation rapidly changed with the advent of airships and airplanes. Since then photogrammetry using aerial photography has been established as the preferred method for the mapping of large areas.

Pushed by the growing need for airborne images the technical design of the cameras was continuously refined. The first prototype of an aerial camera for serial photography was already presented in 1915 [2]. Further on, the quality of camera lenses and the image format was continuously increased to obtain larger terrain coverage per image. The cameras were initially used hand-held, which was less optimal for the layout of the image block formed from several overlapping images. Thus, later airborne sensors were fixed to aircrafts with special camera holes in their body to realize a vertical viewing direction. High performance and geometrical stable roll film material substituted glass plates. Refined and efficient techniques for image data recording were introduced. They comprise high quality optical systems with extremely high resolving power, forward motion compensation, stabilized platform mount, photo flight navigation and aircraft guidance, as well as direct measurement of sensor's exterior orientation during flights.

Scientific Fundamentals

The acquisition of imagery is based on the principles of photography. Photography is a passive method, i. e. the energy, reflected from the object is recorded by photo sensitive material or elements. Until recently this was exclusively done using analogue films. They are now increasingly replaced by digital sensor elements. Consequently, today in operational photogrammetric environments analogue as well as digital sensors are employed.

The benefits of direct digital image recording in comparison to the former digitization of analogue imagery via scanning are obvious: There are cost and time savings because analogue films and film development is not necessary any longer. The time consuming film scanning is dispensable. Besides this, digital recorded images provide better radiometric quality. This positively influences the later automated point measurements in photogrammetric processing. Digital sensors allow for a parallel acquisition of pan-chromatic and multi-spectral image data opening up new fields of application.

Traditionally, photogrammetry is concerned with three-dimensional object reconstruction from two-dimensional images. Similar to human stereo vision photogrammetric object reconstruction is based on two or more different images from the same object with certain image overlap. This need of sufficient image overlap results in specific image block structures. In terrestrial or close range applications the images are mostly taken all around the object of interest realizing convergent viewing directions. In airborne applications the image acquisition is following a pre-planned regular flight pattern. The images themselves are arranged in flight lines with overlaps within the flight line and between neighbouring lines. For satellite imagery the flight path is determined by the satellite's orbit. Many satellites are able to steer their imaging sensor in cross-track and/or along-track orbit direction to obtain stereo image coverage.

Since photogrammetry focuses on the precise geometric reconstruction, the design of imaging sensors has to follow certain requirements. Within three-dimensional object reconstruction the correct reconstruction of imaging rays is essential, i. e. the determination of the sensor's interior and exterior orientation. In order to obtain the interior orientation of the sensor, a calibration is performed. Until recently such calibration was mainly done in laboratories. Today there is a clear trend towards alternative calibration methods [3].

Particularly, the digital imaging sensors become more complex and heterogeneous. Some of them are using more than one optical component and they are often equipped with additional components such as navigation sensors. Calibration is shifted towards a more system oriented approach. From this view point in-situ calibration provides a powerful tool to calibrate and validate such digital sensors. It is already well established for geometrically less stable imaging sensors, e. g. for terrestrial close-range applications. Here, in contrast to the stable geometry of the traditional mapping cameras changes in sensor geometry over time prevent an a priori laboratory based calibration.

In the remainder the focus is on the sensors used for airborne photogrammetric applications, as this is the by far largest area of photogrammetry. Typically, large image formats are employed in order to guarantee efficient data acquisition. The available image format directly influences the effort to cover a certain area with imagery. Therefore, traditional analogue mapping cameras have been designed for very large format films with standard formats of about $23 \times 23\,\text{cm}^2$. Typically focal lengths of 30 cm (normal angle, yields a field of view (FOV) from corner to corner of approximately 60 deg), 15 cm (wide-angle, FOV approx. 95 deg) and 8 cm (super wide angle, FOV approx. 125 deg) are used to adapt to different application scenarios. Analogue mapping cameras have been manufactured by different system suppliers, while the majority of analogue aerial imagery is taken by only two different mapping cameras, namely the Intergraph Z/I (formerly Zeiss) RMK-Top and the Leica Geosystems (formerly LH-Systems, Wild) RC30 series and their predecessors.

Both cameras are very similar, which in the past has pushed the development of measuring and data evaluation tools, independently from the imaging sensor itself. The major part of the mapping system is the camera body with the camera lens cone including the shutter and several lenses. In addition to its very high resolution the optical part has to fulfil geometric requirements leading to measurement accuracies in the range of a few micrometers. Differently from consumer cameras the mapping camera has to preserve its stable geometry for a long period of time and under changing environmental conditions. The camera is fixed in the aircraft (mostly) using an active mount which isolates the camera from the aircraft vibrations and additionally controls the attitude and heading of the camera. Due to this stabilization the airborne images are taken close to nadir viewing (i. e., horizontally) regularly. On the other hand, the active control of attitude variations minimizes the blurring effects caused by aircraft rotations during image recording. The remaining image blur which is due to the forward component of aircraft movement is typically compensated by shifting the film whilst image exposure. This is called forward motion compensation (FMC).

As an example the Intergraph Z/I RMK Top15 is depicted in Table 1. The camera is fixed in the stabilized mount and on top of the camera body the removable film magazine can be seen.

Almost 2000 large format mapping cameras have been distributed all over the world over more than four decades. As of 2005 around 800 of them are still used in different kinds of operational applications. Nonetheless, the era of analogue imagery in photogrammetry comes to an end, similarly to the development in the consumer market. In May 2006 the camera manufacturer Intergraph Z/I announced that there are currently no plans to manufacture new RMK-Top systems. This proofs that digital sensors can compete with the analogue sensors. The official introduction of commercial digital airborne cameras started in 2000.

In order to obtain large formats in digital imaging, two different sensor designs are used. The first relies on a small number of digital CCD sensor lines which can be offered with a reasonable length. These lines are grouped perpendicular to the aircraft's flight direction. Full terrain coverage is obtained via the aircraft's motion. This line scanner concept is also named pushbroom scanning and is known from satellite imaging. In photogrammetry, the systems are often referred as three-line scanners, although typically more than three lines are used to obtain three panchromatic channels as well as four multi-spectral channels. All CCD lines provide the same number of pixels regularly. Thus pan-chromatic and multi-spectral images are obtained with the same geometric resolution. Since the individual physical placement of the lines within the focal

Photogrammetric Sensors, Table 1 The Intergraph Z/I RMK Top15 analogue mapping camera

Intergraph Z/I RMK Top15
– wide-angle lens, focal length 153 mm
– Angular field of view 93 deg (corner to corner)
– Aperture f/4 – f/22, continuously variable
– Exposure time 1/50s – 1/500s, continuously variable
– Remaining distortions $\leqq 3\mu$m
– Film length 150m with 0.1 mm film thickness
– Gyro stabilized mount with $\pm$5 deg in ω, ϕ, $\pm$6.5 deg in κ
– Weight ~165kg (including mount, magazine and control unit)

plane is different, each CCD line provides a different viewing direction.

Contrary to this, large format frame based sensors combine several individual camera heads, each one equipped with one or more CCD frame sensors. All these camera heads are fixed to one airborne platform. The smaller format images, taken by the separate individual camera heads, provide certain image overlaps. This allows for the generation of one synthetic large format image afterwards. Typically, the large format image is taken in the pan-chromatic channel. Additionally, multi-spectral channels are captured simultaneously by additional camera heads, but typically with less spatial resolution compared to the large format virtual pan image. High-resolution colour imagery is obtained from later processing, where the colour channels are combined with the high-resolution PAN images. In these cases, the original radiometric colour information will be more or less impacted, depending on the algorithm and ratio used. This process is termed pan-sharpening. Similar concepts are used in satellite imaging.

In general the frame based digital sensors try to transfer the classical concept of photogrammetric 2D image data processing from the analogue to the digital world, whereas the line scanning approach is following an alternative concept with its long image strips compared to the individual image frames from frame sensors.

Many of the digital airborne sensors are combined with additional sensors for the direct measurement of exterior orientation elements at the time of exposure. High performance integrated GPS/inertial systems are available to solve this task [4,5]. If the GPS/inertial exterior orientation elements are obtained with sufficient accuracy, the photogrammetric image orientation can be done without any additional ground control (so-called direct georeferencing) [6]. In case of pushbroom scanners the use of GPS/inertial components is inevitable. Due to their less stable image geometry (only 1-dimensional lines are recorded instead of 2-dimensional image frames) the additional GPS/inertial measurements are necessary to compensate for the image distortions caused by the sensor's movement during image data acquisition.

The following Table 2 briefly summarizes the main characteristics of three commercially available large format digital sensors. Two of them – namely DMC from Intergraph Z/I [7] and Ultracam-X from Microsoft (formerly Vexcel) [8] – are following the frame concept, whereas the Leica Geosystems (formerly LH-Systems) sensor ADS40 [9] is one representative of the pushbroom line scanners. Besides these, other digital sensors are available and new ones are emerging.

ADS40 and DMC were officially introduced to the market in 2000, whereas the Ultracam-D was presented in spring 2003. Already in 2006 modifications of Ultracam-D and ADS40, namely Ultracam-X and ADS40 (2nd generation) were presented to the photogrammetric community. Meanwhile, more than 100 systems altogether have been sold (status 2006), with the digital mapping sensor market quite equally distributed between all three sensors. System manufactures expect future system sales of ~15–20 individual systems per year. Restricted to the before mentioned three different sensors, this will result in an annual increase of 45–60 digital systems per year. Within another 5 years period the number of available large format digital sensors will be about 325–400 at least.

In addition to that, other camera formats with small to medium image sizes are also used in airborne applications [10,11]. Those systems are based on the frame sensor concept and now (2007) provide images about 4000 × 5500 pix or 7000 × 5500 pix. They were usually not designed for the use in airborne photogrammetric environments originally and, therefore, cannot fully compete with the large format sensors in terms of accuracy, image quali-

Photogrammetric Sensors, Table 2 Overview on today's commercial large format digital airborne mapping cameras

Sensor	DMC	Ultracam-X	ADS40, 2nd generation
Manufacturer	Intergraph Z/I	Microsoft/Vexcel	Leica Geosystems
# camera heads	4 + 4	4 + 4	1
Focal length PAN	120 mm	100 mm	62.5 mm
MS	25 mm	33 mm	62.5 mm
Image size PAN	13824 × 7680 pix	14430 × 9420 pixel	12000 × 1 pix staggered (nadir view)
MS	3000 × 2000 pix	4992 × 3328 pixel	12000 × 1 pix
Pan sharpening (for colour imaging)	yes, applied	yes, applied	no (original spectral bands)
Physical pixel size	12 μm	7.2 μm	6.5 μm
Field of View			
across-track	69 deg (PAN)	55 deg (PAN)	64 deg
along-track	42 deg (PAN)	37 deg (PAN)	
Optics	Carl-Zeiss Jena, system specific	Linos/Vexcel, system specific	Leica, system specific

ty and efficient coverage, especially for large area projects. Nevertheless, due to their reduced physical size and higher flexibility these systems can be accommodated in small aircrafts or even unmanned airborne vehicles (UAV). They are well suited for local use, e. g., repeated flights for monitoring or for quick response for disasters. It is interesting to note, that the number of small to medium format digital airborne cameras is by far exceeding the number of large format cameras today. In many cases they are not only used stand-alone but integrated with other sensors like laser scanners and direct georeferencing components.

Key Applications

Photogrammetry in its original sense is defined as the discipline of the quantitative analysis of photographs. The focus is on the reconstruction of three dimensional object geometry from two dimensional images. Traditionally the main field was in mapping.

The key applications are highly correlated with the source of sensor data and are divided in satellite, airborne, and terrestrial / close range applications. For satellite images the focus is on large area coverage and accuracies of 1 meter and less. For airborne photogrammetric tasks the areas covered are less extended, but typically geometric accuracies in the range of centimetres to few decimetres are required. The highest accuracies are required for close range applications, like highly detailed reconstruction of architectural sites or in industrial or medical environments. Especially for the later additional sensors besides cameras are used. Here maximum accuracies in the range of submillimeter to micrometer are realized.

Future Directions

As illustrated above, the world of photogrammetry is split at the moment: New digital sensors are used parallel to well established analogue cameras. Nevertheless, the future of photogrammetric data acquisition will be fully digital – this is already almost the case for the processing of data.

Although the focus in the sections above was almost exclusively on airborne cameras, the world is more heterogeneous! Besides optical cameras (covering different image formats) other sensors based on laserscanning (LIDAR) or radar technology are used. In terrestrial and close range applications additional measuring devices such as stripe or pattern projectors are in use. In addition to pure geometric reconstruction multi-spectral data recording and analysis becomes more and more important. A clear trend towards multi-sensor systems is obvious. Traditional platform carriers such as aircrafts, helicopters and satellites are supplemented by new carriers like remotely controlled or autonomous flying model aircrafts, helicopters and airships for low flying altitudes. On the other hand, high altitude long endurance (HALE) UAVs allow for data recording from very high altitudes of more than 20 km. Pushed by various novel sensors and platforms new application areas, e. g. monitoring, disaster mapping, precision farming, real estate and tourism evolve. These applications have in many cases less stringent geometric accuracy requirements.

New applications require new concepts for system design and data processing. For example, for monitoring information is needed with a (very) high update rate but confined to a limited area of interest (i. e. daily monitoring of traffic flows in a city). As a consequence a fast and fully automatic processing of data is needed. Fully automatic image matching and feature extraction is highly desirable. In future these tasks might be solved directly on the sensor's chip. This ultimately could lead to intelligent sensor systems or sensor networks.

Cross References

- ► Photogrammetric Applications
- ► Photogrammetric Products

Recommended Reading

1. Meyer, R., Meydenbauer, A.: Baukunst in historischen Photographien. Fotokinoverlag, Leipzig (1985) ISBN: 3731100630, 259 pages
2. Albertz, J.: 90 Jahre Deutsche Gesellschaft für Photogrammetrie und Fernerkundung e.V. Photogrammetrie-Fernerkundung-Geoinformation PFG 5(1999), pp. 293–349 (1985)
3. Cramer, M.: EuroSDR network on digital camera calibration. IAPRS, Vol. XXXV, Part B, Istanbul, digitally on CD-Rom, (2004) 6 pages
4. Mostafa, M., Hutton, J., Reid, B.: GPS/IMU products – the Applanix approach. Fritsch/Spiller (eds.) Photogrammetric Week 2001, pp. 63–83, Wichmann Verlag, Heidelberg, Germany (2001)
5. Kremer, J.: CCNS and AEROcontrol: Products for efficient photogrammetric data collection. Fritsch/Spiller (eds.) Photogrammetric Week 2001, Wichmann Verlag, Heidelberg, Germany, pp. 85–92. (2001)
6. Schwarz, K.-P., Chapman, M.A., Cannon, M.E., Gong, P.: An integrated INS/GPS approach to the georeferencing of remotely sensed data. Photogr. Eng. Remote Sens. PE & RS, **59**(11), 1667–1674 (1993)
7. Hinz, A., Dörstel, C., Heier, H.: Digital Modular Camera: System concept and data processing workflow. IAPRS, vol. XXXIII, part B1, Amsterdam, pp. 164–171, digitally on CD-Rom. (2000)
8. Leberl, F., Gruber, M., Ponticelli, Bernoegger, S., Perko, R.: The Ultracam large format aerial digital camera system. Proceedings ASPRS 2003 annual conference, Anchorage, Alaska, May 2003, digitally on CD-Rom (2003) 7 pages
9. Sandau, R., Braunecker, B., Driescher, H., Eckart, A., Hilbert, S., Hutton, J., Kirchhofer, W., Lithopoulos, E., Reulke, R., Wicki, S.: Design principles of the LH Systems ADS40 airborne digital sensor. IAPRS, vol. XXXIII, part B1, Amsterdam, pp. 258–265, digitally on CD-Rom (2000)
10. Mostafa, M.: Design and performance of the DSS. Fritsch (ed.) Photogrammetric Week 2003, Wichmann Verlag, Heidelberg, Germany, pp. 77–87, 2003. Also published in Proceedings International Workshop on Theory, Technology and Realities of Inertial/GPS Sensor Orientation 22.–23. September 2003, Castelldefels, Spain, digitally on CD-Rom (2003)
11. Cramer, M.: Performance of medium format aerial sensor systems, IAPRS, vol. XXXV, part B, Istanbul, digitally on CD-Rom (2004) 6 pages

Photogrammetry

- ► Data Acquisition, Automation
- ► Intergraph: Real Time Operational Geospatial Applications
- ► Photogrammetric Applications

Photo-Textured

- ► Photogrammetric Products

Pixel

- ► Spatial Data Transfer Standard (SDTS)

Pixel-Based Prediction

- ► Bayesian Spatial Regression for Multi-source Predictive Mapping

Pixel Size

- ► Spatial Resolution

Plane Sweep Algorithm

JORDAN WOOD[1], SANGHO KIM[2]
[1] Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA
[2] ESRI, Redlands, CA, USA

Synonyms

Sweep line algorithm; Spatial join

Definition

The plane sweep (or sweep line) algorithm is a basic computational geometry algorithm for finding intersecting line segments. The algorithm can run in $O(n \lg n)$ time, where n is the number of line segments. This algorithm can be altered to solve many related computational-geometry problems, such as finding intersecting polygons. In spatial databases, we are generally looking at the special case of finding the intersections of minimum bounding rectangles (MBR, see definitional entry).

Historical Background

The plane sweep algorithm, or sweep line algorithm, originated from line segment intersection problem in computational geometry field. The paper and thesis written by Michael Shamos in the middle of 1970 first addressed the computational geometry problems. Later the book [1] written by Preparata and Shamos in 1985 contributed to making people widely aware of the problems. The plane sweep algorithm is one of the main topics in the book, along with other subjects such as convex hull, Voronoi diagram, and all-line-intersections. The plane sweep algorithm has been actively studied since then and expanded rapidly with numerous journal articles. Many application areas have also started using the algorithm. In robotics and motion sensing, it is critical to detect when any two objects intersect for collision. In computer graphics, the ray shooting method requires determination of the intersection of ray with other objects. In spatial databases, the spatial join algorithm deploys the idea of plane sweep algorithm as described in this article.

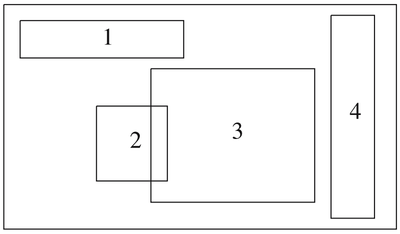
Scientific Fundamentals

The most common use of the plane sweep algorithm in spatial databases is to perform a spatial join. A spatial join is used to answer questions such as, "list all census blocks within 5 miles of an international airport." If we are dealing with a large dataset, such as a map of the US which contains all census blocks and all airports, this could be a very computationally intensive query. The naïve approach to this problem would be to create a set of 5 mile radius circles centered at the airports, then compare the polygon of each census block to see if it overlaps with each of the circles. This would be a very inefficient algorithm. Several different optimizations could be applied to this scheme. First, a spatial indexing scheme could be applied to organize the data so that only census blocks physically near an airport would be tested for overlap. This is an important optimization for the plane sweep algorithm, since it assumes that all the lines or polygons to be analyzed will fit in main memory. Also, a filter and refine strategy decreases the computation time for finding the intersection of complex objects, while increasing the number of objects which can be held in main memory at one time.

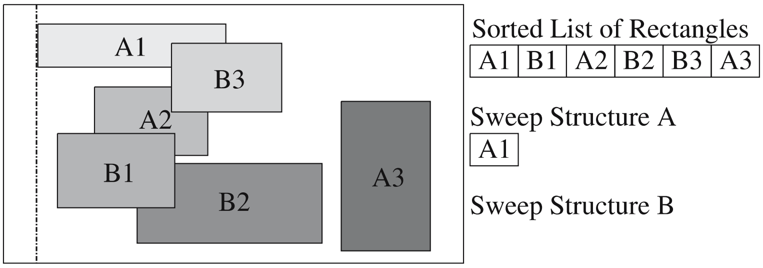
The problem of finding the intersecting rectangles from two sets of rectangles where exactly one rectangle comes from each set is interesting from a spatial database perspective, because the filter step of many spatial joins can be reduced to this problem. First, the geometric shapes are simplified to their minimum bounding rectangles (MBR). The filter step consists of finding the intersection of these MBRs. Then in the refine step, only those shapes whose MBRs have intersected are themselves tested for intersection. Orenstein [2] demonstrated the first use of the filter and refine technique for spatial joins using MBRs. A plane sweep algorithm is commonly used to find the intersection of the two sets.

Algorithm Description

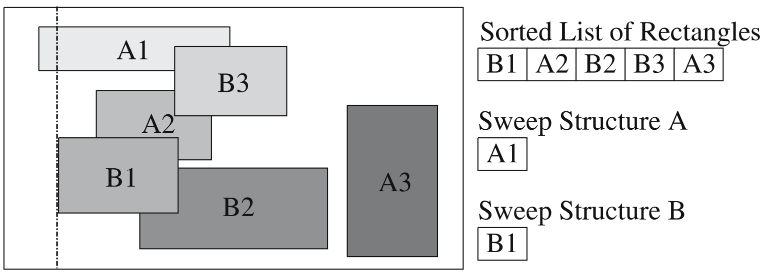
The plane sweep algorithm finds all overlapping MBRs from two sets. There are two phases to this process. First, the rectangles from both sets are sorted in increasing order based on their left sides. Figure 1 shows an example of this ordering. In the second phase, a vertical scan line is swept from the left to the right, stopping at each left rectangle side. All rectangles which are crossed by the scan line as it sweeps across the input are considered 'active'. Only active rectangles need to be tested for intersection.



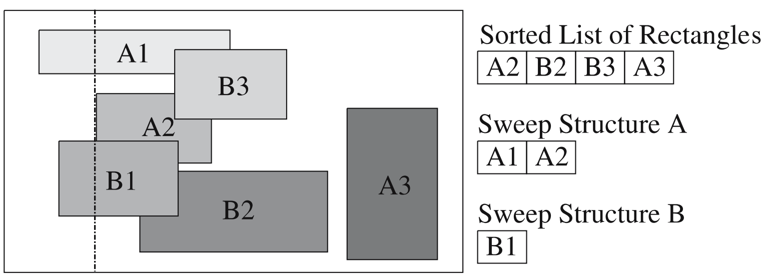
Plane Sweep Algorithm, Figure 1 A set of MBRs



Plane Sweep Algorithm, Figure 2 First iteration of the algorithm



Plane Sweep Algorithm, Figure 3 Second iteration of the algorithm



Plane Sweep Algorithm, Figure 4 Third iteration of the algorithm

Every time the sweep line stops, all inactive rectangles are removed, we test for intersection with the new rectangle, and we add the new rectangle to the active list. A data structure called a sweep structure, which can support the addition, removal, and intersection operations in a time efficient manner, should be used to store the active list. For example, a dynamic interval tree (which is a Cartesian tree) can be used.

Spatial joins require that we take rectangles which are partitioned into two sets, A and B, and find all the pairs of rectangles which overlap, where one rectangle is from set A and the other is from set B. We must therefore have a sweep structure for each of the sets. When the list of rectangles is sorted in the first phase of the algorithm, we must merge the two sets into a single list, while keeping track of which set the rectangle is a member of. As we consider each rectangle in the merged list, we test for intersection with the active rectangles which are members of the opposing set.

Example of Algorithm Execution

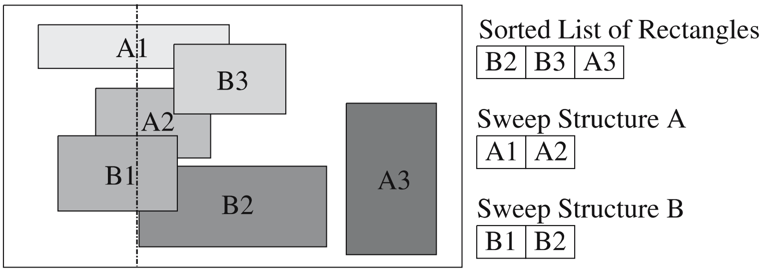
We will use the following example to illustrate the basic plane sweep algorithm used in the filter step of a spatial join. Set A has three members, as does set B. The rectangles are stored as the coordinates of their lower left and upper right coordinates (e. g., (A1.xl, A1.yl) and (A1.xu, A1.yu)). We use these values to determine whether a given pair of rectangles intersect, and whether a given rectangle is to the left of the sweep line. The sweep line is shown as a dashed line. When the sweep line reaches a rectangle from set A, we examine only the members of set B for intersection, and vice versa.

Figure 2 shows the first iteration of the algorithm. There are no elements currently in sweep structure B, so A1 cannot intersect with any of the members of B. At the end of this iteration, we remove A1 from the head of the sorted list.

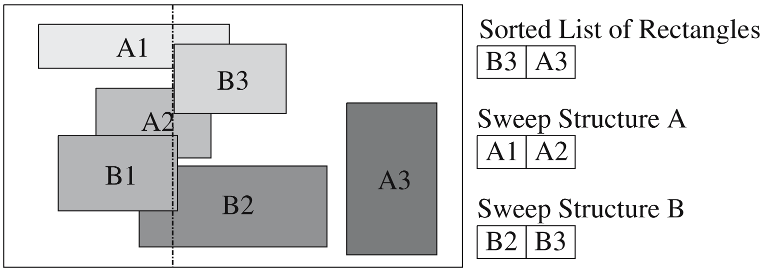
The second iteration is shown in Fig. 3. We compare the y-values of A1 and all the elements of sweep structure B (in this case just B1) to see if they intersect. They do not, and there are no intersecting pairs found in this iteration.

Figure 4 displays the third iteration. When we compare A2 with the one element in sweep structure B, we find that they do intersect. This iteration therefore produces the intersecting pair A2:B1.

The fourth iteration is shown in Fig. 5. We compare B2 to the elements of sweep structure A, and find no intersecting pairs.



Plane Sweep Algorithm, Figure 5 Fourth iteration of the algorithm



Plane Sweep Algorithm, Figure 6 Fifth iteration of the algorithm

The fifth iteration is shown in Fig. 6. We have removed B1 from sweep structure B because it is no longer crossed by the sweep line. This is not strictly necessary, but it keeps the sweep structures from continually growing.We compare B3 with the elements of sweep structure A, and we produce the intersecting pairs B3:A1 and B3:A2.
In the sixth and last iteration, sweep structure B is empty, so no intersections are produced.

Key Applications

The most common use of the plane sweep algorithm in spatial databases is to perform a spatial join. A spatial join is used to answer questions such as, "list all census blocks within 5 miles of an international airport."

Future Directions

The maximum number of rectangles which the sweep line intersects at any point can be referred to as the maximum overlap. This is an important characteristic for determining the performance of the algorithm. Many GIS datasets have highly skewed data where a large percentage of the data is clustered into small areas. This can greatly increase the maximum overlap.When working with very, very large datasets; the amount of memory available for the storage of the active rectangles can become larger than the available physical memory, which can cause a degradation in performance. In this case, we can either reduce the number of rectangles being compared by integrating spatial indexes (such as R-trees) into our spatial join, or by using more sophisticated adaptations of the plane sweep algorithm.

Cross References

► Filter and Refine Strategy
► Minimum Bounding Rectangle

Recommended Reading

1. Baorzsaonyi, S., Kossmann, D., Stocker K.: The Skyline Operator. ICDE (2001)
2. Stojmenovic, I., Miyakawa, M.: An optimal parallel algorithm for solving the Maximal Elements Problem in the Plane. Parallel Computing (1988)
3. Matousek, J.: Computing dominance in E^n. Information processing letters (1991)
4. Roussopoulos, N., Kelley, S., Vincent, F.: Nearest Neighbour Queries. SIGMOD (1995)
5. Hjaltason, G., Samet, H.: Distance Browsing in Spatial Databases. ACM TODS (1999)
6. Kossmann, D., Ramsak, F., Rost, S.: Shooting Stars in the Sky: An Online algorithm for Skyline Queries. In Proceedings of VLDB'02, pp. 275–286 (2002)
7. Papadias, D., Tao, Y., Fu, G., Seeger, B.: An optimal and progressive algorithm for Skyline Queries. ACM SIGMOD (2003)
8. Chomicki, J., Godfrey, P., Gryz, J., Liang D.: Skyline with Presorting. In proceedings of ICDE'03, pp. 717–816, IEEE Computer Society (2003)
9. Lin, X., Yuan, Y., Wang, W., Lu, H.: Stabbing the Sky: Efficient Skyline Computation over Sliding Windows. In Proceedings of ICDE'05, pp. 502–513. IEEE Comp. Soc. (2005)
10. Papadias, D., Tao, Y., Fu, G., Seeger, B.: Progressive Skyline Computation in Database Systems. ACM Trans. Database Syst. 30(1):41–82 (2005)
11. Tan, K.L., Eng, P.K., Ooi, B.C.: Efficient Progressive Skyline Computation. In Proceedings of VLDB'01, pp. 301–310 (2001)
12. Preparata, F.P., Shamos, M.I.: Computational Geometry – An Introduction. Springer, Berlin, Heidelberg, Germany (1985)

Planning Support Systems

► Geocollaboration

Plausibility

► Computing Fitness of Use of Geospatial Datasets

PM-Quadtree

► Quadtree and Octree

Point, Conjecture

► Vague Spatial Data Types

Point, Kernel

► Vague Spatial Data Types

Point Nearest-Neighbor Query

► Nearest Neighbor Query

Point Patterns

► Data Analysis, Spatial

Point Query

► R*-tree

Point, Vague

► Vague Spatial Data Types

Pointless Topology

► Mereotopology

Point-Quadtree

► Quadtree and Octree

Polynomial Spatial Constraint Databases

BART KUIJPERS
Theoretical Computer Science Group, Hasselt University and Transnational University of Limburg, Diepenbeek, Belgium

Definition

The framework of constraint databases provides a rather general model for spatial databases [4]. In the constraint model, a *polynomial spatial constraint database* contains a finite number of relations, that, although conceptually viewed as possibly infinite sets of points in some real space $\mathbf{R}^n$, are represented as a finite union of systems of polynomial equations and inequalities.

Main Text

More specifically, in a *polynomial spatial constraint database*, a relation is defined as a boolean combination (union, intersection, complement) of subsets of some real space $\mathbf{R}^n$ (in applications, typically $n = 2$ or 3) that are definable by polynomial constraints of the form $p(x_1, \ldots, x_n) \geq 0$, where p is a polynomial in the real variables $x_1, \ldots, x_n$ with integer coefficients. For example, the spatial relation consisting of the set of points on the upper half of the unit disk in $\mathbf{R}^2$ can be represented by the formula $x^2 + y^2 < q1 \wedge y \geq 0$. In practice, spatial relations will occur extended with thematic alpha-numeric information, like a name. In mathematical terminology, these spatial relations are known as *semi-algebraic* sets and their properties have been studied extensively [1].

Historical Background

The polynomial constraint database model was introduced by Kanellakis, Kuper, and Revesz [2] in 1990. The application of this model to spatial databases was described by Paredaens, Van den Bussche, Van Gucht [4]. This model was studied extensively in the 1990s and a state of the art book "Constraint databases," edited by G. Kuper, L. Libkin, J. Paredaens appeared in 2000 [3] and the textbook "Introduction to Constraint Databases" by P. Revesz was published in 2002 [5].

Cross References

► Constraint Database Queries
► Constraint Databases and Data Interpolation
► Constraint Databases and Moving Objects
► Constraint Databases, Spatial
► Indexing Spatial Constraint Databases

P

► MLPQ Spatial Constraint Database System
► Visualization of Spatial Constraint Databases

Recommended Reading

1. Bochnak, J., Coste, M., Roy, M.-F.: Géométrie algébrique réelle. Springer-Verlag, Berlin/Heidelberg (1987)
2. Kanellakis, P.C., Kuper, G., Revesz, P.Z.: Constraint query languages. J. Comp. Syst. Sci. **51**, 26–52 (1995)
3. Kuper, G.M., Libkin, L., Paredaens, J. (eds.): Constraint Databases. Springer-Verlag, Berlin/Heidelberg (2000)
4. Paredaens, J., Van den Bussche, J., Van Gucht, D.: Towards a theory of spatial database queries. In: Proceedings of the 13th ACM Symposium on Principles of Database Systems, pp. 279–288 (1994)
5. Revesz, R.Z.: Introduction to Constraint Databases. Springer-Verlag, New York (2002)

Polynomials

► Biomedical Data Mining, Spatial

Polynomials, Orthogonal

► Biomedical Data Mining, Spatial

Populating, Topology

► Oracle Spatial, Geometries

Population Distribution During the Day

BUDHENDRA BHADURI[1]
Geographic Information Science and Technology Group, Oak Ridge National Laboratory, Oak Ridge, TN, USA

Synonyms

Daytime population; Mobile population; Nonresidential population

Definition

Population distribution during the day can be defined as the distribution of population in an area during the daytime hours. However, a precise definition of daytime hours is challenging, given the geographic variability in the length of a day or daylight hours. The US Census Bureau used "normal business hours" as the span of time to describe daytime population [1]. Given that censuses typically estimate residential population, it represents a nighttime population distribution. In that respect, the daytime population in an area may be broadly defined as distribution of population at times other than when they are expected to be at their residences at night which extends the duration from business hours to include the evening hours as well.

Historical Background

Population data has served as a fundamental backbone for planning sustainable development. There is evidence from the early 1600s of a population census in Virginia where people were counted in nearly all of the British colonies that became the United States at the time of the Revolutionary War [2]. Historically, it has been used to meet a variety of long term socioeconomic and political planning needs. For example, the first US Census of 1790, which counted 3.9 million residents, helped raise the membership in the US House of Representatives from an original 65 to 105. Modern censuses that include not only population count, but also its demographic and socioeconomic characteristics have had a tremendous impact on many aspects of our society covering, among other things, urban planning and housing development, transportation planning, energy demand and infrastructure planning, healthcare planning, environmental impact assessment, emergency preparedness and response, and scientific research. However, the majority of these planning activities have been aimed at medium- to long-term solutions over a number of years and consequently a general geographic assessment of population, described through their residential locations, was adequate to address such planning processes. Movement of population during a day results directly from people traveling to the locations of their daytime activities (employment, business, educational institutions, and recreational locations), away from their residences [3]. The patterns of such population displacements depend on the relative geographic distribution of residential and business areas. In most modern societies, these two activity locations are distinctly separated in space, and employment or business locations contain fewer residences than businesses. Consequently, a large number of people move into these areas while only a few leave, resulting in a substantial swelling in the daytime population of that area. The motivation to formalize the concept of nonresidential and daytime population distribution is rooted predominantly in two areas. First, it is widely understood that analysis of the daytime population distribution provides a very competitive economic advantage, as business-

[1]The submitted manuscript has been authored by a contractor of the U.S. Government under contract DE-AC05-96OR22464. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

es are enabled to target specific consumer bases depending on their locations and convenience of access during that majority of the 24-h period when people are out of their residences. In recent years, a stronger requirement for understanding daytime population has emerged from the emergency preparedness and response community to assess the at-risk population from the threats of technological and natural disasters and deliberate attacks on human lives such as terrorist events.

Scientific Fundamentals

Population movement is a function of both geographic space and time. The mobility of a population during the day is driven by people's need to temporarily relocate for activities such as education (schools, colleges, universities), employment, businesses (shopping, post offices, restaurants, and others), or recreation (parks, museums, other tourist attractions). In general, the daytime population distribution of an area can be conceptually expressed as:

$$\begin{aligned}\text{Daytime Population} &= \text{Workers} + \text{School children}\\ &\quad + \text{Tourists} + \text{Business travelers}\\ &\quad + \text{Residual Nighttime Residential Population}\end{aligned}$$

or,

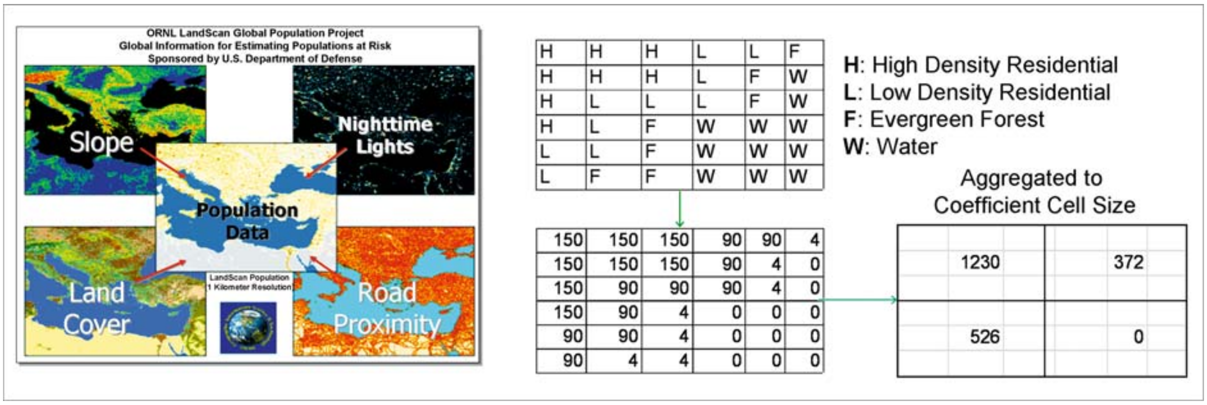
$$\begin{aligned}\text{Daytime Population} &= \text{Nighttime Population}\\ &\quad + \text{Daytime incoming population}\\ &\quad - \text{Daytime outgoing population}\end{aligned}$$

Deriving a quantitative estimate from the above qualitative expressions involves further analyses of population data, which can be represented as:

$$\begin{aligned}&\text{Daytime Population} =\\ &\quad \text{Nighttime Residential Population}\\ &\quad - \text{Workers leaving during the day}\\ &\quad + \text{Workers moving in during the day}\\ &\quad - \text{School children leaving during the day}\\ &\quad + \text{School children moving induring the day}\\ &\quad + \text{Tourists visiting during the day}\\ &\quad + \text{Businesstravelers coming into the area}\end{aligned}$$

Although this may not be the most accurate representation, it is largely accepted as the expression leading to the best available daytime population estimates. It follows that the development of quantitative estimates of daytime population distribution involves two distinct components. The first component involves identification of daytime activity locations such as businesses, schools, and other recreational activities. The second component covers the identification and distribution of the mobile population that are at those locations. Usually, it is easier to gather information on the first as these are static geographic features and are commonly captured in public and commercial databases for various infrastructures, or can be derived from remote-sensing-based land-cover data, high-resolution satellite and aerial photographs, or state and local government data. However, it is extremely challenging to obtain information on the number and nature of movement of people during the day that comprehensively captures the net displacement of the nighttime residential population during daytime. Although, detailed population movement data sets may be available for isolated local communities, they are not available at a national scale. In fact, the US Census Bureau's compilation of journey-to-work data is the only readily available and nationally consistent data set for the US that describes people's movement from residences to employment locations. Consequently, the US Census Bureau's estimate of daytime population based on the 2000 Census only reflects populations based on travel to work. Similarly, it does not limit the work-related commuting to specific hours. All worker-related travel, irrespective of what time of the day it occurs, has been used to derive these estimates of daytime population [1].

An important aspect of daytime population distribution is the geospatial scale at which it is estimated. Theoretically, the finest spatial resolution achievable through the map algebra technique described above is directly tied to the finest scale of the available input data. For example, the US Census Bureau collects worker commuting data at the census tract level and reports national daytime population distribution at the county level. It also reports estimates of daytime population for key cities in each state. Similar city level estimates of daytime population from government and commercial sources are available for Japan [4], Canada, and the US. All these data sets appear to be heavily focused on worker population movement during the day and the data is presented through vector data models (points and polygons). For example, daytime population fluxes are restricted to individual county and city boundary polygons. Some commercial databases represent individual activity locations as points that potentially offer high spatial accuracy but mostly account for worker population at individual business locations. In reality, the data sets necessary to comprehensively estimate daytime population exist in the forms of points and polygons, which makes it challenging to create a high-resolution population distribution through simple map algebra analysis. It requires integration of disparate spatial data and advanced geospatial modeling where the spatial mod-



Population Distribution During the Day, Figure 1 Example illustrating the use of land-cover data in the LandScan dasymetric model

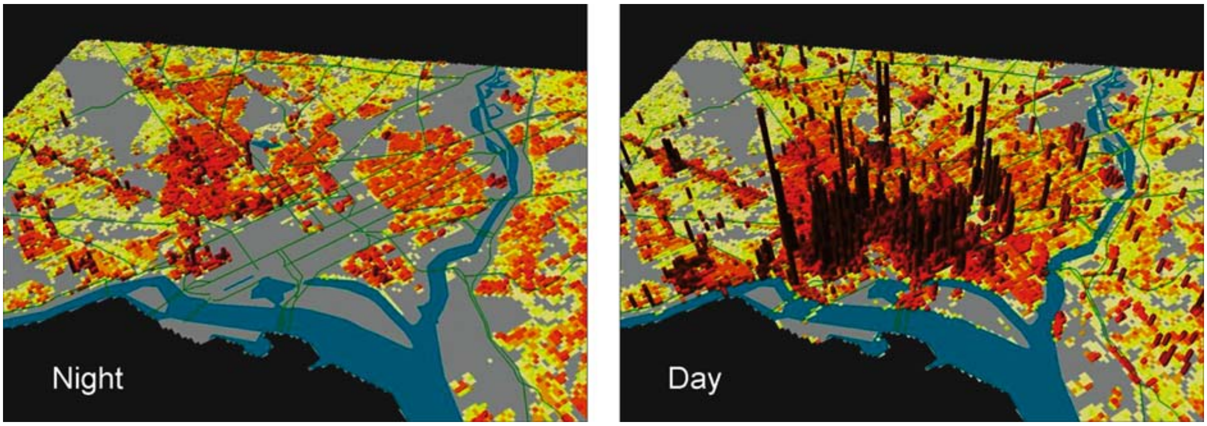
el enables decomposition of the input data into finer spatial resolutions and then representation through uniform raster or gridded datasets.

Decomposition of population distribution estimates has been a well-known problem. Several interpolation and decomposition methods have been developed to address this issue with census (polygonal) population data. They include areal weighting, pycnophylactic interpolation, dasymetric mapping, and various smart interpolation techniques. Areally weighted interpolation is the simplest of the methods, where a regular grid is intersected with the Census polygon and each grid cell is assigned a value based on the proportion of the polygon contained in each cell [5,6,7]. This method implies an assumption of uniform distribution of population, which is not a realistic solution for decomposition of population data. Pycnophylactic interpolation extends the areal weighting methodology by iteratively applying a smoothing function to the raster cell values, with the weighted average of its nearest neighbors, while preserving the total population count of the polygon [8]. This method creates a continuous surface which contradicts the obvious discontinuous nature of population distribution. Dasymetric modeling is analogous to areal interpolation but uses ancillary spatial data to aid in the interpolation process. The ancillary spatial data is at a finer spatial resolution and the variability in its values enables an asymmetric allocation of population values. Land cover/land use is the best example in this respect, where different land cover or land use categories for each cell can be used as weighting functions for population distribution such that urban areas will have a higher weight than forested areas (Fig. 1) [7,9,10]. Smart interpolation, in principle, is a multidimensional version of a dasymetric model where the allocation refinement comes from more than one ancillary data sources, which are at a finer resolution than the population polygon [10,11]. The utility of such interpolation techniques at local scales are well documented. In fact, there are two well-known publicly available data sets, the Gridded Population of the World (GPW) [12] and the LandScan Global Population database [13], which employ such a method to produce global population distribution data. While GPW is a product of simple areal weighting interpolation at 2.5 arc-min or approximately 5-km cell size, LandScan, at 30 arc-s or approximately 1-km cell size is the finest resolution global population data available to date, derived through advanced spatial data integration and multidimensional dasymetric modeling or smart interpolation.

Although both GPW and LandScan datasets are developed using Census information, GPW depicts a nighttime residential population (i. e., directly decomposed Census data) while the LandScan database represents an “ambient” or average of the 24-h period population (because the model assigns some parts of the populations based on nonresidential activities). Both databases are publicly available for noncommercial usage from Columbia University [14] and Oak Ridge National Laboratory [15] respectively. GPW is updated periodically, while LandScan Global Population database has been updated and released annually since 2000.

The development of daytime population distribution models and databases is significantly more challenging, as it requires further integration and modeling of activity-based datasets into the residential population distribution model. In 2004, the US Census Bureau released the following three daytime population distribution data tables based on the 2000 census [1]:

- Table 1. Leading Places on Percent Change in Daytime Population, by Size (202 highly populated cities)
- Table 2. The United States, States, Counties, Puerto Rico and Municipalities
- Table 3. Selected Places by State (6524 communities)



Population Distribution During the Day, Figure 2 Difference in population distribution between nighttime and daytime as illustrated by LandScan USA data for Washington DC

However, these data sets only take into account the commuting worker population in an area. The best spatial resolution of these data is still at the community level (small cities) and thus is appropriate for general-purpose planning. Expanding on their LandScan Global Population research program, since the early 2000s, the US Department of Energy's Oak Ridge National Laboratory has played a pioneering role in developing an advanced scalable daytime population distribution model for the nation called LandScan USA [16,17]. At an unprecedented resolution of 3 arc-second or approximately 90-m cell size, LandScan USA demonstrates a consistent methodology for developing daytime population distribution (Fig. 2). This enhancement in resolution comes from the incorporation of a large number of high-resolution ancillary data sets used in the LandScan USA dasymetric model. Some of these ancillary data sets include:

- Population
  - Census block population
  - Census tract-to-track worker flow
  - Bureau of Labor Statistics quarterly updates for worker population.
- Roads
  - Tele Atlas North America (formerly known as Geographic Data Technology) Dynamap
  - US Census TIGER data
  - Navigational Technologies (NAVTEQ) roads
- Land Cover/Land Use
  - National Land Cover Data (NLCD)
  - State and local level GIS databases
- Slope
  - National Elevation Data (NED)
- Academic Institutions
  - Department of Education
  - Environmental Systems Research Institute(ESRI)
  - Tele Atlas North America;
- Prisons
  - Department of Justice National Jail Census
- Hospitals
  - American Hospital Association (AHA)
- Business Employment
  - ESRI Business Database (Info USA)
- Ortho Imagery
  - Google Earth
  - Earth Viewer
  - Microsoft Terra Server

In addition to the worker population in an area, LandScan USA database accounts for children of K–12 school age, university students, institutional population (jails and prisons), and a mobile daytime residential population at various activity locations such as shopping malls, post offices, cultural attractions, and recreational facilities (parks). Development of LandScan USA version 1.0 has been completed for the 50 US States and Puerto Rico but the data is not publicly available yet. It should be noted that both LandScan Global and LandScan USA are evolving databases and new ancillary input data sets are continuously added to the model as they become available.

A similar approach [18] has also been adopted for estimating daytime population at a lower resolution of 250-m grid cells. The coarser resolution of this data set has been attributed to coarser resolution of input variables in the model such as county-to-county worker mobility data from Census as compared to tract-to-tract worker mobility data used in LandScan USA. Moreover, this estimation is solely based on worker and residential populations and does not account for populations at academic institutions, commercial retail locations, and recreational areas.

Key Applications

Missions of national priority ranging from socio-environmental studies to homeland security utilize population data as one of the critical elements. High-resolution population distribution data during daytime hours is even more significant for successfully addressing research and practical applications that require an estimation of mobile population. Applications of daytime population distribution are numerous and can be broadly divided into two categories.

Estimating Population at Risk from Disasters

Large numbers of human lives are at risk from natural and technological disasters. Volcanic eruptions, earthquakes, hurricanes, floods, wildfires, blizzards, droughts, and tornadoes are examples of natural disasters that are slow in their onset, predictable in most cases (except earthquakes and volcanic eruptions), and typically geographically restricted. However, natural disasters are unpreventable and for the most part uncontrollable. Technological disasters, on the other hand, occur suddenly (i. e., are unpredictable) but can be controlled and their impact minimized through effective disaster planning and management. Examples of technological disasters include explosions, electrical blackouts, nuclear accidents, and bioterrorism. Critical application domains for estimating the population at risk include national and homeland security, where improved knowledge of where people live relative to sites of potential terrorist activities can refine estimates of potential populations exposed [18] or injured for rapid risk assessment; and emergency preparedness and response where the daytime population can support emergency response resource planning, emergency evacuation planning, and disaster relief delivery.

Public Health and Socioeconomic Analysis

Public health is probably one of the most promising areas that can take great advantage of daytime population distribution information. Disease epidemiology with short- and long-term exposure assessment and evaluating access to health care facilities and locating future health care facilities can be done effectively with an understanding of daytime population distribution. Assessment of the mobility daytime population with respect to their residences also facilitates understanding of contagious disease propagation patterns. Exposure and risk assessment from environmental pollutants for work-related activities can be performed very effectively using the daytime distribution of population. Given that workers are likely to spend 50% or less of their time at home during work days, the daytime population distribution provides a tremendous advantage for occupational exposure analysis over using traditional census data. High-resolution daytime population data also reduces population distribution errors around point and area sources and can be of significant help for environmental justice analysis. Other socioeconomic applications include demographic analysis to evaluate socioeconomic disparity patterns of a region in terms of work-related commuting patterns for different demographic groups and estimating rates and trends of urban sprawl.

Future Directions

Accurate estimation and representation of daytime population distribution poses significant challenges. First, any region witnesses an influx of tourists (or visitors) and business travelers during the day, particularly in large urban areas and cultural/natural attractions (such as national parks). In addition, a large number of people travel along roads driving through major urban areas. Such transitional population is not effectively captured in any consistent and organized databases and will require advanced data integration and modeling techniques to be effectively included in a daytime population distribution. Current spatial modeling and population distribution techniques only locate population at specific activity locations and do not account for the commuting time when the mobile population is on the transportation networks. Thus, a true average daytime population distribution requires details of worker commuting patterns and non-worker travel habits, which includes data for the number of people at necessary service locations such as post offices, banks, shops, and parks. Another important aspect of daytime population distribution is to characterize the temporal variability. The nature of daytime population distribution can be significantly different depending upon whether the data represents a working or a weekend day or holiday. Moreover, there is also a seasonal and weather impact on the daytime population distribution. Summer days do not have students in academic institutions and have more people at outdoor locations. In contrast, a larger population tends to be indoors during days with weather extremes (such as extreme heat, cold, or storms). Thus assessment of a true "representative" daytime population of a region will require development of an average distribution from such different daytime population distribution scenarios.

Cross References

- ► Computing Fitness of Use of Geospatial Datasets
- ► Data Analysis, Spatial
- ► Geodemographic Segmentation
- ► Geographic Dynamics, Visualization And Modeling
- ► Homeland Security and Spatial Data Mining

► Intelligence, Geospatial
► Movement Patterns in Spatio-temporal Data

Recommended Reading

1. US Census Bureau: Census 2000 PHC-T-40. Estimated Daytime Population and Employment-Residence Ratios: 2000; Technical Notes on the Estimated Daytime Population. http://www.census.gov/population/www/socdemo/daytime/daytimepoptechnotes.html (2000). Accessed 17 Sept 2007
2. US Census Bureau: US Census Bureau's Fact Finder for the Nation: History and Organization. http://www.census.gov/prod/2000pubs/cff-4.pdf (2000). Accessed 17 Sept 2007
3. Quinn, J.: The daytime population of the central business district of Chicago. Review by Breese, G.W. Am. Sociol. Rev.**15**, 827–828 (1950)
4. Japanese Statistics Bureau, Ministry of Internal Affairs and Communications: Population Census, Daytime Population http://www.stat.go.jp/english/data/kokusei/2000/jutsu1/00/01.htm (2000). Accessed 17 Sept 2007
5. Goodchild, M., Anselin, L., Deichmann, U.: A Framework for the Aerial Interpolation of Socioeconomic Data. Environ. Planning A **25**, 383–397 (1993)
6. Goodchild, M., Lam, N.: Aerial interpolation: a variant of the traditional spatial problem. Geo-Processing **1**, 297–312 (1980)
7. Mennis, J.: Generating surface models of population using dasymetric mapping. Prof. Geogr. **55**, 31–42 (2003)
8. Tobler, W.: Smooth pycnophylactic interpolation for geographical regions. J. Am. Stat. Assoc. **74**, 519–530 (1979)
9. Wright, J.: A method of mapping densities of population: with Cape Cod as an example. Geogr. Rev. **26**, 103–110 (1936)
10. Langford, M., Unwin, D.: Generating and mapping population density surfaces within a geographical information system. Cartogr J. **31**, 21–26 (1994)
11. Cohen J., Small, C.: Hypsographic demography: the distribution of human population by altitude. Proc. Natl. Acad. Sci. **95**, 14009–14014 (1998)
12. Deichmann, U., Balk, D., Yetman, G. Transforming Population Data for Interdisciplinary Usages: from Census to Grid. http://sedac.ciesin.columbia.edu/gpw-v2/GPWdocumentation.pdf (2001). Accessed 17 Sept 2007
13. Dobson, J., Bright, E., Coleman, P., Durfee, R., Worley, B.: LandScan: a global population database for estimating populations at risk. Photogramm. Eng. Remote Sensing **66**, 849–857 (2000)
14. Center for International Earth Science Information Network: Gridded Population of the World (GPWv3). Columbia University, New York. http://sedac.ciesin.org/gpw/ (2007). Accessed 17 Sept 2007
15. Oak Ridge National Laboratory: LandScan Global Population Project. Oak Ridge National Laboratory, Tennessee. http://www.ornl.gov/sci/landscan/ (2007). Accessed 17 Sept 2007
16. Bhaduri, B., Bright, E., Coleman, P., Dobson, J.: LandScan: locating people is what matters. Geoinformatics **5**, 34–37 (2002)
17. Cai, Q., Rushton, G., Bhaduri, B., Bright, E., Coleman, P.: Estimating small-area populations by age and sex using spatial interpolation and statistical inference methods. Trans. GIS **10**, 577–598 (2006)
18. McPherson, T., Brown, M.: Estimating daytime and nighttime population distributions in U.S. cities for emergency response activities. In: Proceedings of the Symposium on Planning, Nowcasting, and Forecasting in the Urban Zone, 84th AMS Annual Meeting, Seattle WA, 11–15 Jan 2004
19. Dobson, J., Bright, E., Coleman, P., Bhaduri, B.: LandScan2000: A New Global Population Geography, Remotely-Sensed Cities. Taylor and Francis, London, UK (2003)

Position, Absolute

► Indoor Localization

Position Location

► Indoor Positioning with WirelessLocal Area Networks (WLAN)

Position, Relative

► Indoor Localization

Positional Accuracy Improvement (PAI)

CARSTEN RÖNSDORF
Ordnance Survey, Southampton, UK

Synonyms

PAI; Synchronization of spatial data; Map quality; Map overhaul; Digitization of maps; Relative positional accuracy; Absolute positional accuracy; Rubber sheeting; Geometric fidelity; Shifting geometries; Conflation

Definition

Positional Accuracy Improvement is the process of improving the position of the coordinates defining the geometry of a feature in a geospatial dataset to better reflect its "true" position. This "true" position can either relate to the absolute position in an overarching Coordinate Reference System such as WGS-84 or ETRS-89 or to the relative position in relation to the geometry of another feature in the vicinity.

The PAI process is commonly utilized in two different, but related ways:

PAI of Reference Data

This process deals with improving the position of geometries in a reference dataset that describes physical or abstract features of the earth. These reference datasets are typically large-scale cadastral or topographic datasets issued by National Mapping Organizations but can

P

also consist of other regional or local datasets, usually described as "land bases" and used to reference other information.

PAI of User Data

If a geospatial vector dataset is derived from a reference dataset either by digitizing or copying geometries, both datasets have a topological relationship in the sense that the former is based on the latter. In regards to user datasets PAI describes the subsequent synchronization of one or more of the user datasets with the already positionally improved reference dataset in order to re-instate the relationships between geometries.

Historical Background

Mapping as a technology and means of communication about entities of the real world has a history that can be traced back to at least 3500 BC [1]. The quality of a map is directly related to the methods of surveying and cartography used at the time when the map was produced. It is common in surveying and mapping to avoid a complete recollection of the data every time a new map of a previously surveyed area is created. It is good practice to use old maps as a basis and update and enhance them with new and additional information. This technique, which can be described as "map overhaul", is used to create new themes as well as updating a map to a more current status.

Many geospatial datasets that are used in today's digital environments are updated in the same way. In fact, many of these datasets were originally digitized from paper maps and subsequently updated. This means that many geospatial datasets in use today are an amalgamation of data from different sources, integrated at different times by different methodologies.

With the move from paper maps to geospatial datasets the ability to easily combine spatial data from different sources has dramatically increased. This allows various independently created datasets to be jointly presented and analyzed. The spatial integration of data from various sources requires an understanding about the positional accuracies of the geometries in the datasets to avoid mismatches and misinterpretations. Positional accuracy is an important quality aspect of geospatial datasets, a viewpoint that is underpinned in the data quality description in ISO19113 [2].

To give an example: A dataset digitized from a small-scale map will have a radically different accuracy than another one surveyed by differential GPS. Geospatial reference datasets, particularly large-scale topographic and cadastral data issued by Local or National Mapping Organizations, were typically created over the course of decades or centuries. Varying accuracies have their origin in different surveying and transformation methods that were applied to the data after their initial creation.

Positional Accuracy Improvement as a term was initially introduced by Ordnance Survey of Great Britain to describe its coordinated program to improve the positional accuracy of large scale topographic data [3]. This program was one of the major activities to obtain a large-scale reference dataset to express the topography of the whole of Great Britain and is used as a case study later in this chapter.

A series of subsequent publications and workshops [4,5] have established the term PAI to articulate systematic changes to improve the position of geospatial reference data and its effect on datasets that were derived from these. It was found that the issues that lead to PAI may vary in different national contexts, but that the core technical problem is often identical [4].

Scientific Fundamentals

The positional accuracy of geodata can be described by the terms relative and absolute accuracy, which are defined as follows.

Relative Positional Accuracy, which has traditionally been used to indicate the positional accuracy of maps, is defined as

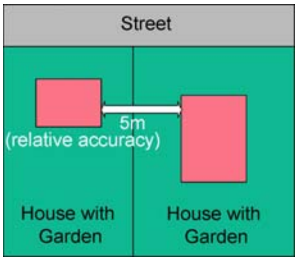
> the difference of the distance between two defined points in a geospatial dataset and the true distance between these points within the overall reference system.

Practically, the true distance can be measured using conventional terrestrial surveying techniques, such as a tape or laser distance measure, and can be compared to the calculated length between the two data points. An example for relative positional accuracy is given in Fig. 1.

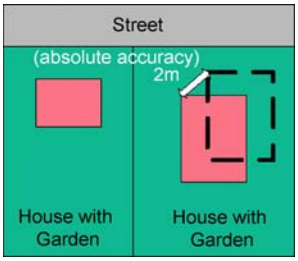
In the 1980s satellite navigation technology, such as GPS, introduced the possibility of obtaining a point's coordinate directly without relating to neighboring features. Therefore another accuracy definition is needed.

Absolute Positional Accuracy is defined as

> the distance between a defined point in a geospatial dataset and its true position in the overall reference system.



Positional Accuracy Improvement (PAI), Figure 1 Relative positional accuracy



Positional Accuracy Improvement (PAI), Figure 2 Absolute positional accuracy

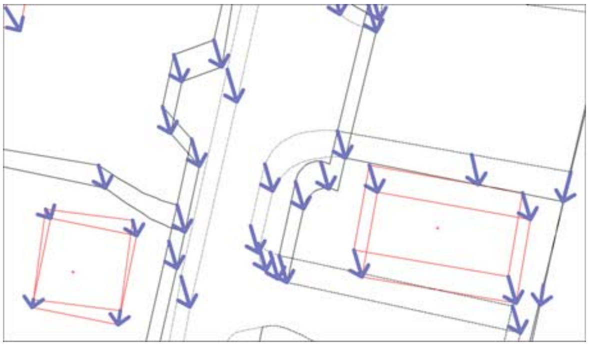
Practically, the true position within the reference system can be determined to centimeter accuracy using differential GPS surveying. An example for absolute positional accuracy is given in Fig. 2.

Both the absolute and relative positional accuracy of a given dataset can be determined by calculating the differences between the mentioned distances of a significant sample of the dataset and can be expressed as a root mean square error (RMSE) that relates to a one sigma (standard deviation) confidence level. Assuming a Gaussian distribution for the errors or differences, this means that the likelihood that a measurement falls into the range expressed by the RMSE—plus or minus one meter, for example—is 67%.

Historically, relative accuracy has been more important to most users than absolute accuracy since it expresses a local quality statement of a geospatial dataset taking the relationship of neighboring features into account.

Topological Relationships Between Geometries

Vectors between identical points of geometries in an unimproved and an improved (or shifted) datasets are called **Link Vectors**. The start and end-points of these vectors have coordinate values in the overall reference system. This means that they describe the shift between the points in both datasets. A **Link Vector Field** is a collection of link vectors for a dataset or part of a dataset. An example for a link vector field connecting identical points in two displaced sets of geometries is displayed in Fig. 3.



Positional Accuracy Improvement (PAI), Figure 3 Link vector field

The process of altering the coordinate values of points in a datasets by adding link vectors to individual coordinates is called **Rubber Sheeting**. If the link vector field does not contain a link vector that originates on the point to be shifted, a link vector is interpolated within the link vector field. Commonly used interpolation algorithms include nearest neighbor, inverse distance weighting or natural neighbor [6].

The term **Geometric Fidelity** is used to describe how well the shape of a line or area geometry is retained in a PAI process [7,8]. Geometric fidelity is defined as

> the difference between the shape of a geometry or set of geometries as an ordered sequence of points in a geospatial dataset and the sequence of the corresponding points in the real world.

Practically, the geometric fidelity can be assessed for each geometry either by visually checking the distortion of a geometry or by calculating and comparing the angle between edges in the dataset and the real world. The latter can be approximated by GPS measurements of the end points of the lines.

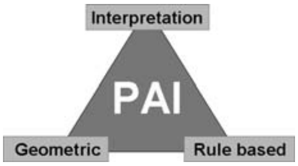
The process of rubber sheeting will decrease the geometric fidelity of a dataset in a PAI process in most cases. The geometric fidelity will not be altered if the link vector field leads to a close approximation of a linear shift and or a rotation.

Shifting Geometries

In addition to re-surveying or re-capturing data, there are three fundamental ways to move data in a PAI scenario. They are illustrated in Fig. 4, the **Shifting Triangle**.

Shifting by Interpretation describes the method of moving features to their improved position by determining this position on an individual basis for every geometry. It is based on the perceived relationship between these features and others in the vicinity. The new position is determined by a human operator or, alternatively, an artificial intelligence process, by interpreting the feature according to an intangible or tangible capture specification.

Geometric shifting relates to the application of a link vector field that describes the difference between the old and the improved base data by means of geometric transformations. Rubber sheeting is a commonly used example for geometric shifts.



Positional Accuracy Improvement (PAI), Figure 4 Shifting triangle

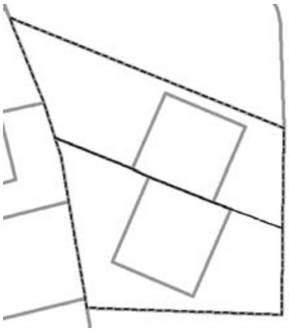
Rule based shifting make use of the specification of the dataset and explicit relationships between features in the base and user datasets. These can be expressed by geometric constraints, such as preserving right angles between lines or calculating distance functions between user and reference geometries [9]. The distance function is described in a later paragraph.

The shifting triangle implies that the most efficient process to migrate user data is likely to be a combination of the three fundamental methods. An often used combination would be to rubber sheet the data first, then 'snap' relevant features back to the improved reference dataset according to a rule, followed by an interpretational correction of geometries that were incorrectly shifted.

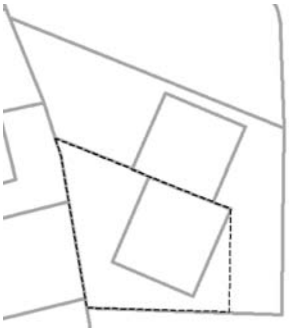
Topological Relationships Between Geometries

The following three figures illustrate important relationships between reference data (thick grey lines) and user data (thin dashed lines) using polygon geometries as an example.

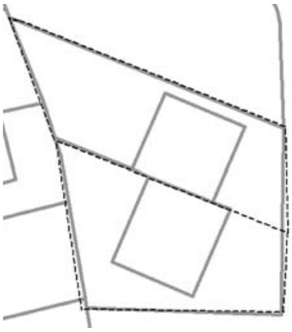
In Fig. 5 the user data completely follow the geometry of the reference data – the vertices of the user data are identical to the vertices of the reference data. Within a Geographic Information System many of this data are created by digitization using a 'snapping' algorithm. Hence these data can be described as 'snapped' user data. In Fig. 6 the user data contain geometry that is not in the reference. Therefore this part of the polygon can't be linked to any geometry in the reference data. Figure 7 depicts a common scenario where the user data are digitized against the reference data without using a 'snapping' algorithm. In this case the user data vertices are not identical with the reference data vertices, but close to them. Since the quality of digitization could be better (as illustrated in Fig. 5), this relationship can be described as roughly digitized user data.



Positional Accuracy Improvement (PAI), Figure 5 Snapped user data



Positional Accuracy Improvement (PAI), Figure 6 Partly snapped user data

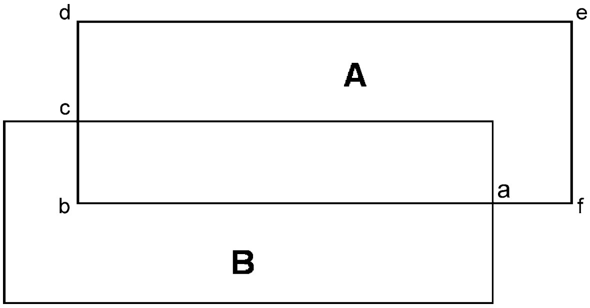


Positional Accuracy Improvement (PAI), Figure 7 Roughly digitized user data

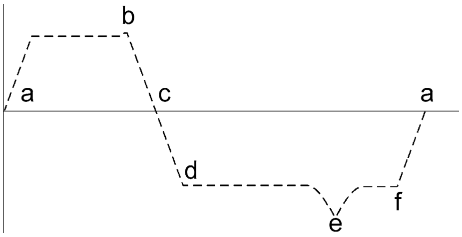
Distance Function

A spatial relationship between two polygons can be described by a distance function [10]. The distance function reports the minimum distance separating two polygons as a function *f(n)* of the perimeter *n* of polygon A. Where two polygons are coincident, the function will report a zero minimum distance. $f(x) > 0$ indicates that the perimeter of polygon A falls within polygon B and therefore overlaps while $f(x) < 0$ shows that the perimeter of polygon A falls outside polygon B.

For the two polygons A and B, as shown in Fig. 8, the distance function between the boundary of polygon A and polygon B is displayed in Fig. 9. It is created by following the perimeter of polygon A from intersection point *a* over *b, c, d, e* and *f* back to point *a*. For all points on the perimeter (or a number of points that are placed in small discrete intervals on the perimeter) the shortest distance to polygon B is calculated.



Positional Accuracy Improvement (PAI), Figure 8 Two simple polygons



Positional Accuracy Improvement (PAI), Figure 9 Distance function expressing the relationship between the polygons in Fig. 8

The distance function for snapped user data (see Fig. 8) constantly equals zero, while the distance function for partly snapped polygons (see Fig. 9) will be zero for the snapped part of the polygon and have a distinctive peak where user data does not follow reference geometries. For the scenario shown in Fig. 7, the roughly digitized data, the distance function will be close to zero, with some noise indicating the difference between the user data and underlying reference data [9].

A term related to PAI is **Conflation**, meaning automated map compilation [11]. This technique, pioneered in 1985, allows the integration of two overlapping geospatial datasets by matching some corresponding structures in both sets and subsequent transformation of all features into one dataset. While conflation is a technique that can be utilized to execute PAI, PAI itself focuses on maintaining the synchronicity between two or more datasets following geometry changes to the reference dataset and allows a number of different technical solutions.

A fundamental discussion of the Mathematical Models for Geometrical Integration that are the basis for PAI as well as links to additional literature can be found in an article by Kampshoff [12].

Key Applications

Positional Accuracy Improvement introduces geometry changes to geospatial datasets in order to improve their relative and/or absolute positional accuracy.

While an improvement of the relative accuracy is a possible scenario, today's main application area is the improvement of the absolute accuracy within a national or global coordinate system. The influence of new surveying methods, namely the impact of Global Navigation Satellite Systems, on the production of large-scale topographic and cadastral datasets has been the main driver to improve the positional accuracy of these datasets. New surveys are often very accurate within a global Coordinate Reference System and need to be integrated into existing topographic or cadastral reference datasets. A large part of the existing data is typically based on historic surveys in local reference systems of a lesser accuracy.

Positional Accuracy Improvement (PAI), Figure 10 Reference and user data

Reference Data and User Data

In practical terms geospatial data can usually be divided into two categories: reference data that provide the geospatial context and user data that comprise additional features supporting a particular application. In many cases the user data are generated by the users themselves while the reference data are provided by a National Mapping Agency or private data provider. On a paper map both the base map and the user data are usually drawn or printed onto the same surface and cannot easily be separated from each other. Digital datasets maintain both as two or more completely separate layers that are just combined for analysis and publication on the screen or in print. An example for reference and user data is given in Fig. 10. The thick, grey lines represent reference data; the thin, red lines indicate user data.

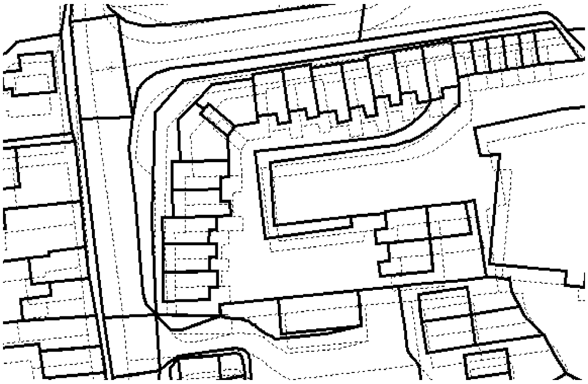
PAI on Reference Data

The improvement of an existing reference dataset can be done by re-surveying the geometries, shifting them or a combination of both. Figure 11 gives an example for the difference between an unimproved (thick line) and an improved (thin, dashed line) reference data set.

After a reference dataset is shifted, all future updates of that dataset will typically apply to the improved dataset. This usually triggers a PAI process on user data in conjunction to datasets that were derived from this reference dataset.

PAI on User Data

User datasets that were derived from or created to be used in conjunction with a particular reference dataset may not



Positional Accuracy Improvement (PAI), Figure 11 Reference data before (*bold line*) and after applying PAI (*dashed line*)

be synchronized with that reference dataset after the latter one has been improved. This means that the topological relationships between geometries in the reference dataset and the user data may be destroyed.

This can have a significant impact on the use of these datasets. Applications such as automated searches for conveyancing, may produce incorrect results if a search is done using improved reference data in conjunction with unimproved user data.

Once a user dataset is shifted to its post-PAI position, geometric interoperability between this dataset and the improved reference is re-established and both datasets can be used in conjunction again.

Use Case: PAI in Great Britain

In Great Britain, the original topographic surveys date as far back as the early 1800s. More importantly, a large amount of the surveys that form the backbone of today's large-scale digital reference data was acquired during the first half of the 20th century. At that time it was common practice to use separate, county-specific reference and coordinate systems to survey and display the maps (known as County Series maps). A fundamental approach to integrate those projections into one common metric coordinate system for Great Britain, the British National Grid, was started around 1938 and finished after the Second World War. Today all large-scale reference data are held in a national geographic database that currently holds about 440 million features as well as selected spatial relations between features.

It was apparent, even before GPS was used as a surveying tool, that new topographic details could not always be seamlessly integrated into the national geographic database, resulting in operational overheads to maintain the database as well as using the data in conjunction with user data.

While differential GPS methods deliver absolute positional accuracies of 10 centimeters or better, features in large-scale Ordnance Survey data have an absolute positional accuracy of between 2.8 m RMSE in rural areas and 0.4 m RMSE in urban areas. This indicates the accuracy of the absolute position of a coordinate in the context of the British National Grid coordinate system. In contrast to this, the relative positional accuracy has always been significantly better.

Following earlier debates that go back to the 1970s, Ordnance Survey started to plan a national program to improve the absolute positional accuracy of its rural large-scale data in the late 1990s. It applies to 152,000 km^2 (or about two thirds of the area of Great Britain) and excludes the major urban areas, which were already resurveyed to a higher standard from 1947 onwards, as well as mountain and moorland regions, where improving the positional accuracy is not necessary or economically viable. The program was completed in March 2006 and is future-proofing the value of the national geographic database. The absolute positional accuracy of the data after the improvement is 1.1 m RMSE in rural areas and 0.4 m RMSE in selected rural towns.

In Great Britain this data set is widely used as a reference in conjunction with individual user datasets by several hundred organizations throughout the country. The need to improve the positional accuracy of associated user datasets has created at least 30 different solutions and services to shift user datasets [13].

Use Case: MAF/TIGER Accuracy Improvement in the United States

In the United States the most prominent PAI program is undertaken by the US Census Bureau as part of the MAF/TIGER Accuracy Improvement Project [14]. The program aims at improving the accuracy of the TIGER (Topologically Integrated Geographic Encoding and Referencing System) database to 3.8 meters RMSE for all 50 states as well as Puerto Rico and the U.S Virgin Islands. Unimproved data have been reported to differ up to 150 meters from its (true) Differential-GPS position. This will allow the Bureau to match geographic locations to census geographies in a more automated way.

Future Directions

It is anticipated that the widespread use of GPS and aerial photography that is rectified against GPS control points will significantly increase the importance of absolute accuracy. Therefore the need to positionally improve datasets with a low positional accuracy is likely to increase in order to make them geometrically interoperable with datasets of a higher absolute positional accuracy.

To date most PAI programs were implemented as coordinated programs. These programs usually focused on the improvement of a particular dataset and were focused on the individual requirements for geometric interoperability of this dataset with those of a higher positional accuracy. In the future the development of a more generic process to improve the accuracy of geospatial datasets would be beneficial. Particularly the implementation of PAI as a web service, performed on the fly, as the data are being prepared or loaded into a client application, will be a powerful tool in order to make datasets geometrically interoperable.

Cross References

This article has been prepared for information purposes only. It is not designed to constitute definitive advice on the topics covered and any reliance placed on the contents of this article is at the sole risk of the reader.

► Data Infrastructure, Spatial

Recommended Reading

1. Dorling, D., Fairbairn D.: Mapping: ways of representing the world. Longman, London (1997)
2. International Organization for Standardization: ISO 19113 Geographic information – Quality principles. Geneva (2002)
3. Havercroft. M.: Consultation paper: Positional accuracy of large-scale data and products. Ordnance Survey, Southampton (1997)
4. Bray, C., Woodsford, P.: Positional Accuracy Improvement: Impacts of improving the positional accuracy of GI databases. In: EuroSDR official workshop publication. no. 48, Dublin. http://www.eurosdr.net/km_pub/no48/html/positional/positional_index.htm (2004)
5. Roensdorf, C.: PAI2: Achieving Geometric Interoperability of Spatial Data. In: EuroSDR official workshop publication. no. 49, Dublin: http://www.eurosdr.net/km_pub/no49/html/PAI2/index.htm (2005)
6. Hettwer, J., Benning, W.: Restklaffenverteilung mit der Natural-Neighbour-Interpolation. Allg. Vermess.-Nachr. **110**, 122–129 (2003) (in German)
7. Chrisman, N.R.: The Error Component in Spatial Data. In: Maguire, D.J., Goodchild, M.F., Rhind, D. W. (eds.) Geographical Information Systems, Longman, London (1991)
8. Goodchild, M. Gopal, S. (eds.): Accuracy of Spatial Databases. Taylor & Francis, London (1989)
9. Roensdorf, C.: Achieving and Maintaining Interoperability of Spatial Data, FIG working week proceedings, Cairo. http://www.fig.net/pub/cairo/papers/ts_14/ts14_02_ronsdorf.pdf (2004)
10. Straub, B.M., Wiedemann, C.: Towards the update of geodata by automatic object extraction. In: Serpico, S.B (ed.) Image and Signal Processing for Remote Sensing VI. SPIE Proceedings vol. 4170, p. 304. The International Society for Optical Engineering Bellingham, WA (2000) http://ipi216.ipi.uni-hannover.de/html/publikationen/2000/straub/straub_gerke_wiedemann.pdf
11. Saalfeld, A.: Conflation: Automated Map Compilation. US Bureau of Census, SRD Research Report Number: Census/SRD/RR-87/24. http://www.census.gov/srd/papers/pdf/rr87-24.pdf (1997)
12. Kampshoff, S.: First International Workshop on Next Generation 3D City Models. Bonn (2005) http://www.gia.rwth-aachen.de/Service/document_download.php?fileTag=kambonn05
13. Ordnance Survey, Positional Accuracy Improvement website, www.ordnancesurvey.co.uk/PAI
14. MAF/Tiger Accuracy Improvement Program, http://www.census.gov/geo/mod/maftiger.html. This article has been prepared for information purposes only. It is not designed to constitute definitive advice on the topics covered and any reliance placed on the contents of this article is at the sole risk of the reader

Position-Aware Technologies

► Location-Aware Technologies

Positioning

► Location-Based Services: Practices and Products

PostGIS

CHRISTIAN STROBL
German Remote Sensing Data Center (DFD),
German Aerospace Center (DLR), Weßling, Germany

P

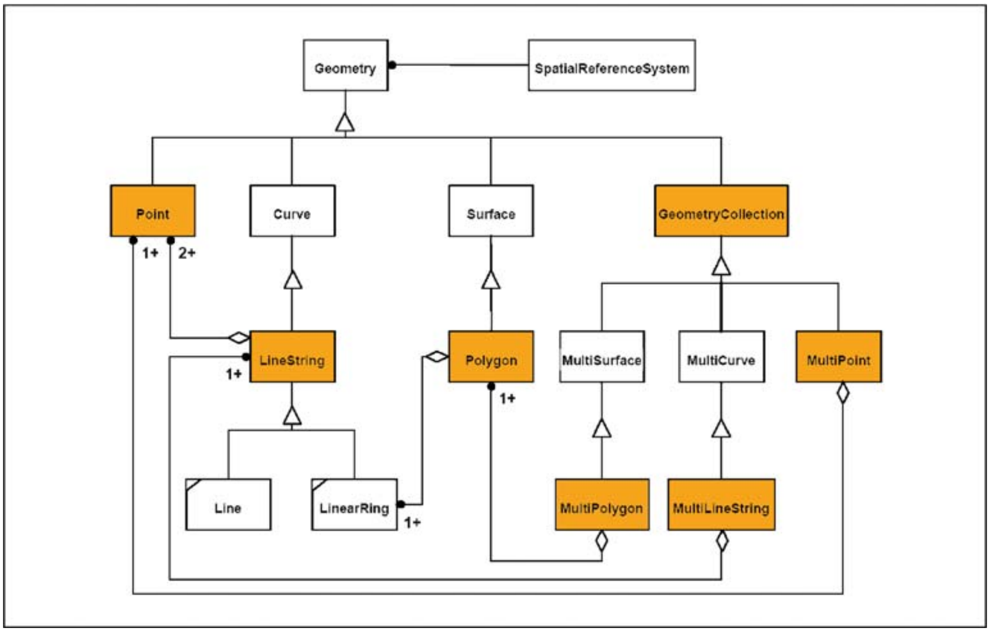
Synonyms

Postgres; OGIS; Spatial DBMS; Public-domain software; Open source; Object-relational; Simple features model; GEOS library; SQL, spatial; R-tree; GiST index

Definition

PostGIS is a spatial database extension for the PostgreSQL (SQL being structured query language) object-relational database. It is certified as a compliant “Simple Features for SQL” database by the Open Geospatial Consortium (OGC).

PostGIS adds geometry data types and spatial functions to the PostgreSQL database. The supported geometry data types are “Points,” “LineStrings,” “Polygons,” “MultiPoints,” “MultiLineStrings,” “MultiPolygons” and “GeometryCollections”. Spatial functions enable the analysis and processing of geographic information systems (GIS) objects. Examples are measurement functions like “Area,” “Distance,” “Length” and “Perimeter” and spatial operators like “Union,” “Difference,” “Symmetric Difference” and “Buffer”. Topological relationships, like “Equals,” “Disjoint,” “Intersects,” “Touches,” “Crosses,” “Within,” “Contains” and “Overlaps”, are processed by the



PostGIS, Figure 1 Geometry class hierarchy of the "Simple Features for SQL" specification from the Open Geospatial Consortium. The geometry types supported by PostGIS are gray shaded

Dimensionally Extended Nine-Intersection Model (DE-9IM).

PostGIS and PostgreSQL are open source. PostGIS is released under the GNU General Public License and PostgreSQL is released under the Berkely Software Distribution (BSD) license.

The functionality of PostGIS is comparable to ESRI ArcSDE, Oracle Spatial, and DB II spatial extender.

Historical Background

The first version of PostGIS was released in 2001 by Refractions Research. It is published under the GNU General Public License [1] and development has continued since then. In 2006, PostGIS was certified as a compliant Simple Features for SQL database by the OGC. It uses libraries of other open source projects. The GEOS (Geometry Engine Open Source) library [2] provides most of the operations described by the OGC Simple Features and the proj4 [3] library contributes the projection support.

Refractions Research is located in Victoria, British Columbia, Canada. It is a consulting and product development organization, specializing in spatial and database application development [4].

The history of PostgreSQL begins at the University of California at Berkeley (UCB). PostgreSQL, originally called Postgres, was created at UCB by a computer science professor named Michael Stonebraker. Stonebraker started Postgres in 1986 as a follow-up project to its predecessor Ingres. Stonebraker and his graduate students actively developed Postgres for 8 years. In 1995, two Ph.D. students from Stonebraker's lab, Andrew Yu and Jolly Chen, replaced Postgres' POSTQUEL query language with an extended subset of SQL. They renamed the system to Postgres95. In 1996, Postgres95 departed from academia and started a new life in the open source world under the BSD license [5]. At the same time the database system was given its current name PostgreSQL. PostgreSQL began at version 6.0 (1996); in 2007 the current version is PostgreSQL 8.2 [6].

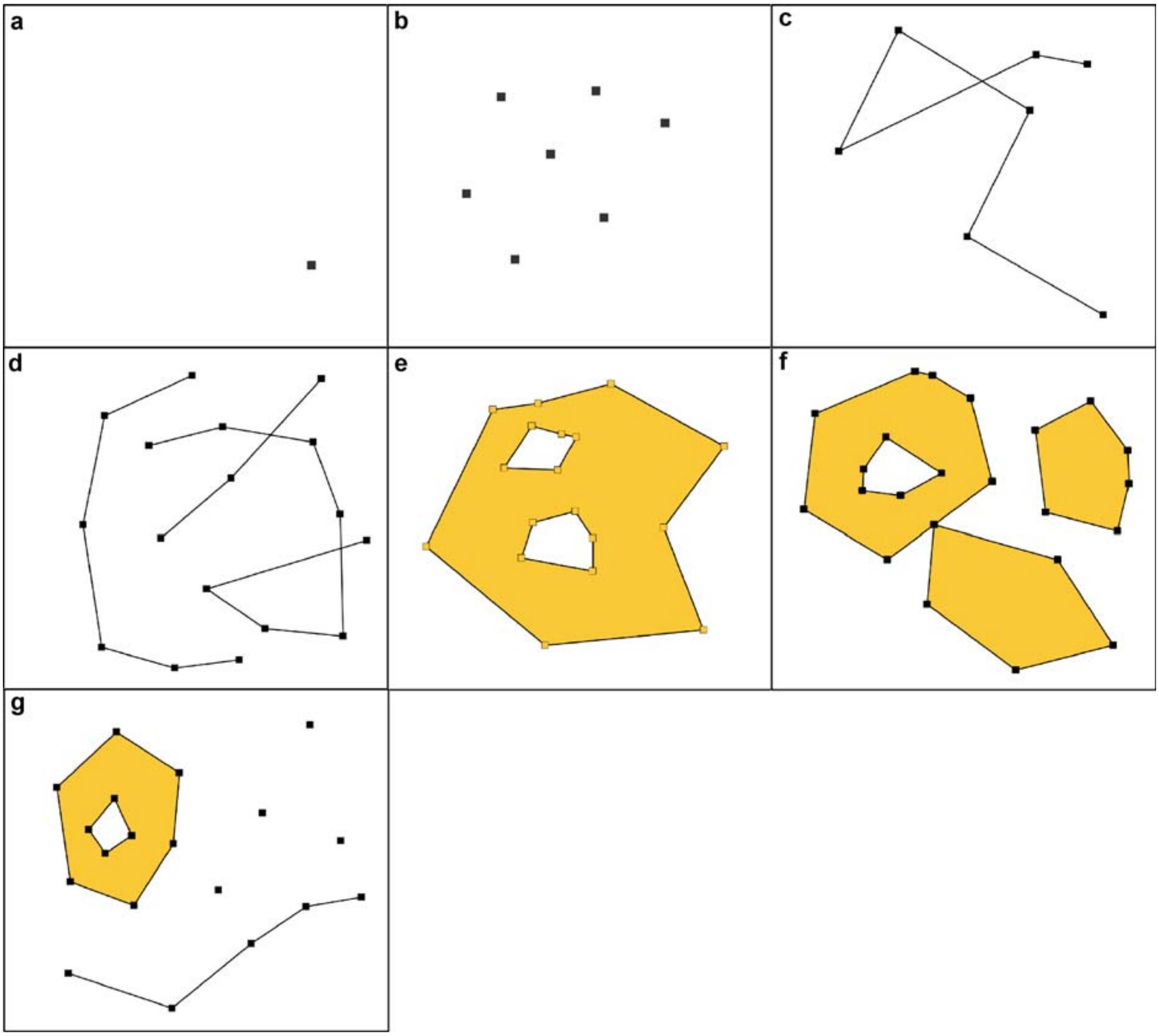
Scientific Fundamentals

Like other spatial databases, PostGIS combines the advantages of classical GIS software, mainly the possibility of spatial analysis, with the advantages of database management systems (DBMS) Such as indexing, transactions and concurrency [7,8].

Simple Features for SQL

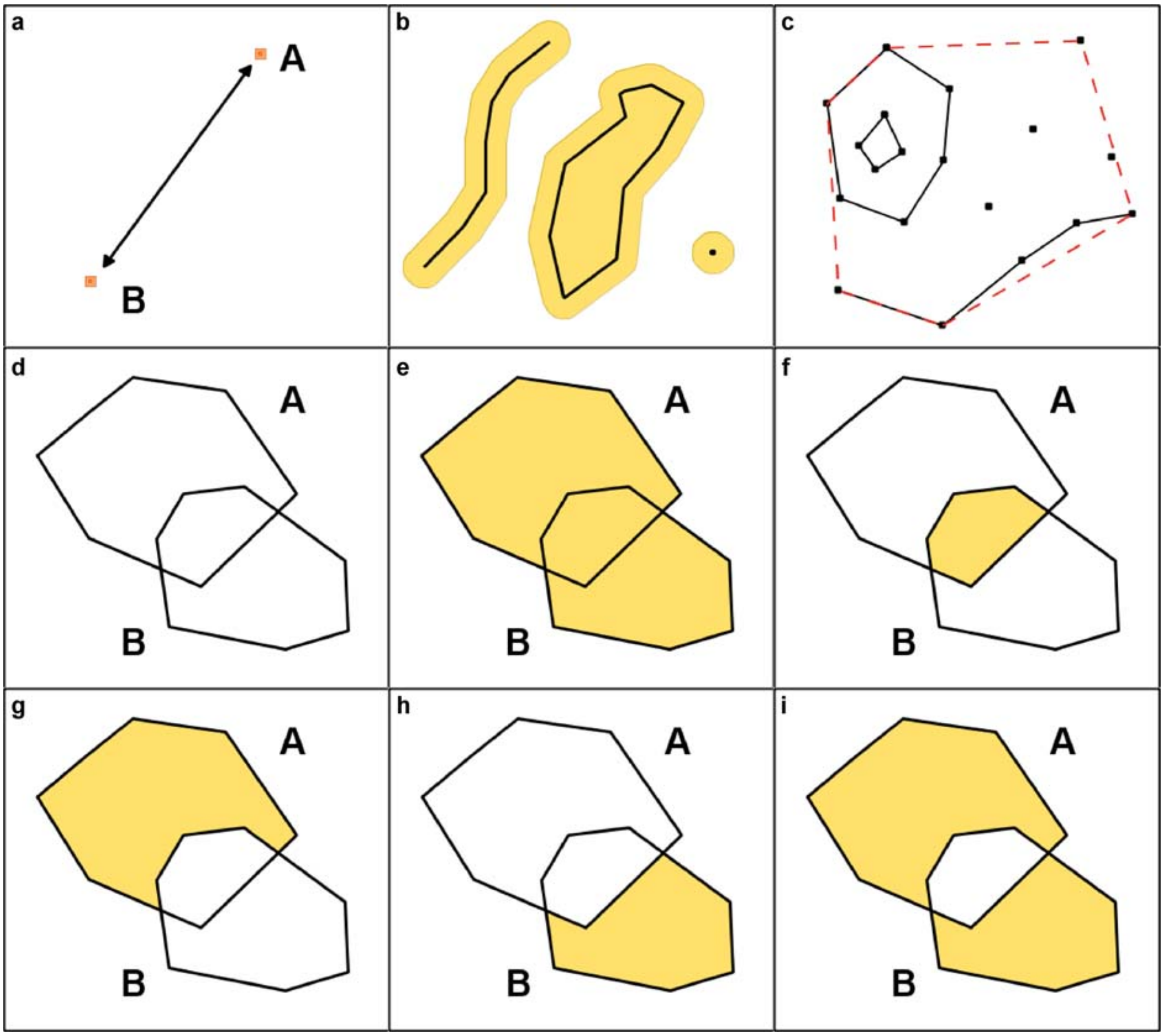
PostGIS follows the Simple Features for SQL specification from the OGC [9]. This implies:

- PostGIS supports the Simple Feature Class Hierarchy according the Open GIS Geometry Model. This includes geometry types for Points, LineStrings, Polygons, MultiPoints, MultiLineStrings, MultiPolygons and GeometryCollections (Fig. 1 and Fig. 2).



PostGIS, Figure 2 Geometry types supported by PostGIS. **a** Point. **b** MultiPoint. **c** LineString. **d** MultiLineString. **e** Polygon. **f** MultiPolygon. **g** GeometryCollection

- PostGIS supports the representation of geometry data as Well Known Text (WKT), Well Known Binary (WKB), as Geography Markup Language (GML) and as Keyhole Markup Language (KML) for Google Earth. Additionally, it supports output as Scalable Vector Graphics (SVG) path geometry.
- PostGIS implements SQL functions that test spatial relationships. These functions include "Equals," "Disjoint," "Intersects," "Touches," "Crosses," "Within," "Contains," "Overlaps" and "Relate". All these operators are based on the DE-9IM [9,10].
- PostGIS implements SQL functions that support spatial analysis. These functions include "Distance," "Buffer," "Convex Hull," "Intersection," "Union," "Difference" and "Symmetric Difference" (Fig. 3).
- PostGIS implements spatial operators for determining geospatial measurements like Area, Distance, Length and Perimeter.
- PostGIS provides information about the geometry type and the spatial reference system. This spatial metadata is stored in the Geometry Columns Metadata View and in the Spatial Reference System Information View according to the Simple Features for SQL specification [9]. Each reference system has a unique identifier called SRID according to the European Petroleum Survey Group (EPSG) code [11].



PostGIS, Figure 3 Spatial functions supported by PostGIS. **a** Distance. **b** Buffer. **c** Convex hull. **e** Union. **f** Intersection. **g** The difference of polygon *A* to polygon *B*. **h** The difference of polygon *B* to polygon *A*. **i** Symmetric difference. **d** The polygons used for the spatial operations of **e**–**i**

Spatial SQL

The implementation of the OGC "Simple Features for SQL" offers GIS new and powerful features for managing, retrieving and analyzing geospatial data. The spatial domain introduces a new set of functions to the SQL Language. The following queries are not complete and give only an elementary review of the potential of the spatial SQL syntax provided by PostGIS:

List the names of all cities which are located inside Bavaria.

```
SELECT city_name
    FROM city a, country b
        WHERE WITHIN (a.geom, b.geom)
        AND b.country_name = 'Bavaria';

        city_name
----------------------------
Munich
Augsburg
...
```

List the names of all countries which are neighbors to Bavaria.

```
SELECT b.country_name
    FROM country a, country b
        WHERE TOUCHES (a.geom, b.geom)
```

```
      AND a.country_name = 'Bavaria';
```

```
        country_name
-----------------------------
Thuringen
Baden-Wurttemberg
Hessen
Sachsen
(4 rows)
```

List the names of all cities which are located within 50 km of the river Isar.

```
SELECT DISTINCT a.city_name
   FROM city a, river b
      WHERE DISTANCE (a.geom, b.geom) < 50000
      AND b.river_name = 'Isar';
```

```
        city_name
----------------------------
Munich
Passau
...
```

Calculate the area of a buffer of 50,000 m around Munich (see also Fig. 3b).

```
SELECT AREA (BUFFER(geom,50.000))/
                        10000 AS Hectares
   FROM city
      WHERE city_name = 'Munich';
```

```
        hectares
------------
780361.288064939
(1 row)
```

List the name, the population and the area of all countries which have an area greater than 3,000,000 ha sorted by the population (in ascending order).

```
SELECT country_name, pop_admin, AREA(geom)/
                        10000 AS Hectares
   FROM country
      WHERE AREA(geom) > 30000000000
      ORDER BY pop_admin;
```

```
country_name        | pop_admin | hectares
--------------------+-----------+-----------------
Niedersachsen       | 8000909   | 4733454.20332757
Baden-Wurttemberg   | 10717419  | 3621827.63163362
Bayern              | 12469000  | 7029553.1603014
Nordrhein-Westfalen | 18058000  | 3438200.2301504
(4 rows)
```

Show the geometry type of the table cities.

```
SELECT DISTINCT GEOMETRYTYPE(geom)
   FROM city;
```

```
geometrytype
------------
POINT
(1 row)
```

Show the description of the spatial reference system for the table countries.

```
SELECT DISTINCT SRID(geom)
   FROM country;
```

```
srid
----
4326
(1 row)
```

With the result of the last query, e. g., the SRID = 4,326, it is possible to get information about the used projection from the table spatial_ref_sys.

```
SELECT srid, proj4text
   FROM spatial_ref_sys
      WHERE srid = 4326;
```

```
srid | proj4text
-----+--------------------------------
4326 | +proj=longlat +ellps=WGS84
                 +datum=WGS84 +no_defs
(1 row)
```


Show the point location of Munich as WKT.

```
SELECT city_name, ASTEXT(geom) AS "Location"
   FROM city
      WHERE city_name = 'Munich';
```

```
city_name | Location
----------+------------------------
Munich    | POINT(11.5429545454545
                     48.1409727272727)
(1 row)
```

Show the point location of Munich as GML, transformed to the coordinate system with EPSG code 31464 (Gauß Krüger, Germany, 12th meridian).

```
SELECT ASGML(TRANSFORM(geom,31464),7)
   FROM city
      WHERE city_name = 'Munich';
```

```
asgml
--------------------------------------
<gml:Point srsName="EPSG:31464">
<gml:coordinates>4466089,5333763
                     </gml:coordinates>
```

```
<gml:Point>
(1 row)
```

Spatial Join (Query Processing)

PostGIS supports spatial joins. A spatial join is comparable to a standard table join based on a spatial relationship. A standard table join merges two tables into one output result. The join is based on a common key.

SELECT a.city_name, b.country_name
FROM city a, country b
WHERE a.country_name = b.country_name
ORDER BY b.country_name, a.city_name;

A spatial join merges two tables into one output result based on a spatial relationship. For example, the names of the countries are stored in the table country and the names of the cities are stored in the table city. If anybody wants to list the name of the cities and the name of the countries, in which the cities are located, in one table, they have to use a spatial join:

SELECT a.city_name, b.country_name
FROM city a, country b
WHERE WITHIN(a.geom, b.geom)
ORDER BY b.country_name, a.city_name;

Indexing and Query Optimization

PostgreSQL supports compound, unique, partial, and functional indexes, which can use any of its B-tree, R-tree, hash, or Generalized Search Tree (GiST) storage methods. GiST indexing is an advanced system, which provides an interface and framework for developers to add their own indexes. It allows the combination of a lot of different sorting and searching algorithms including B-tree, B+-tree, R-tree, partial sum trees, ranked B+-trees and others [6,12,13,14].
PostGIS indexes are R-tree indexes, implemented on top of the general GiST indexing schema. R-trees organize spatial data into nesting rectangles for fast searching ([4,15], Fig. 4).
With PostgreSQL and PostGIS, several possibilities exist for query optimization. It is possible to choose between a sequential scan and an index scan for attribute data and between a sequential scan and an index scan using the GiST index for geometry data.
For mixed spatial/nonspatial queries it is possible to use the index with the best selectivity to provide high-performance query plans.
The spatial indexes are not used automatically for every spatial request or operator. Because the R-tree index is based on rectangles, spatial indexes are only efficient for bounding box comparisons. In PostGIS, the indexed search is activated by using the "&&" operator, which means "bounding boxes overlap". The following SQL statement shows a short example:

SELECT river_name FROM river
WHERE geom && SETSRID('BOX3D
(11 47, 12 49)'::BOX3D,4326)
AND DISTANCE(geom, GEOMFROMTEXT
('POINT(12.0 48.5)', 4326)) < 1;

The example query demonstrates a characteristic "two-step" approach to spatial processing:

- The first step, the so-called filter step, is the indexed bounding box search, which runs on the whole table (geom && BOX3D).
- The second step is the so-called refinement step. It only operates on the filtered subset using the exact geometries and represents the original query, in this case the distance query. As this query runs only on the subset returned by the filter step the high costs of processing the exact feature geometries are minimized.

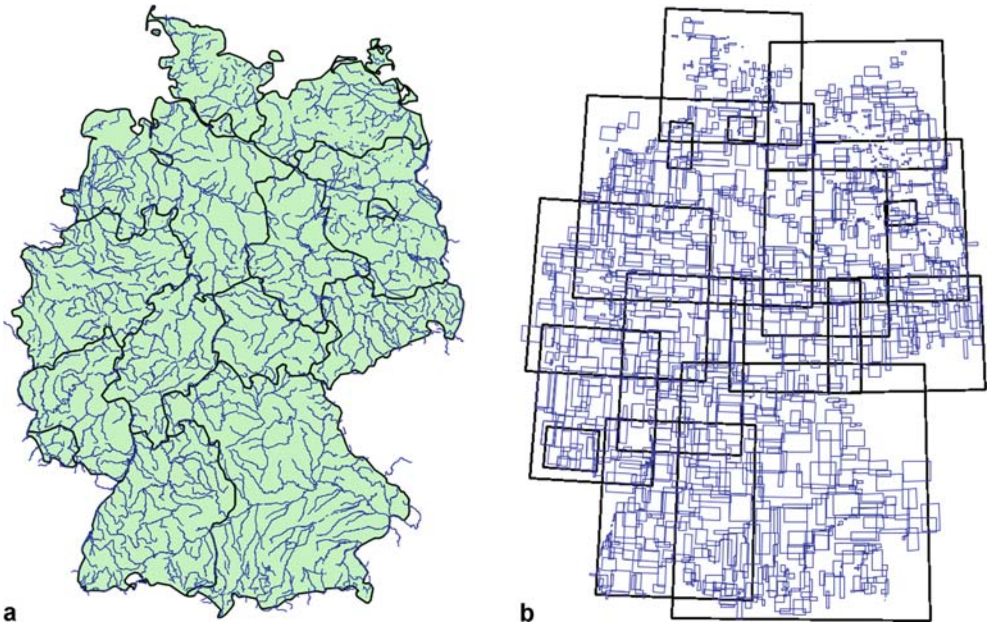
This highly recommended strategy to improve the performance of spatial queries is called the filter-refine paradigm [7].

Key Applications

Spatial data infrastructures (SDIs) facilitate access to geospatial information using a minimum set of standard practices, protocols, and specifications [16]. Every SDI requires a spatial database server and PostGIS represents an open-source- and OGC-compliant solution. Thus PostGIS is supported by many GIS applications, which cover a broad range from server, over workstation and desktop, to internet solutions.

Open Source Software

- deegree: http://www.deegree.org/
- GeoServer: http://geoserver.org/
- GeoTools: http://geotools.codehaus.org/
- GRASS: http://grass.itc.it/
- gvSIG: http://www.gvsig.gva.es/
- MapServer: http://mapserver.gis.umn.edu/
- OGR Simple Feature Library: http://gdal.maptools.org/ogr/
- OpenJUMP: http://openjump.org/wiki/show/HomePage
- Quantum GIS: http://www.qgis.org/
- Thuban: http://thuban.intevation.org/
- uDig: http://udig.refractions.net/
- ...



PostGIS, Figure 4 The bounding boxes that are used for the spatial indexes of the countries and rivers shown in **a** are given in **b**

Proprietary/Closed Software

- ArcGIS (with the Interoperability Extension): http://www.esri.com/
- Cadcorp SIS: http://www.cadcorp.com/
- Feature Manipulation Engine FME: http://www.safe.com/
- Ionic Red Spider: http://www.ionicsoft.com/
- ...

Future Directions

The 1.2.0 release of PostGIS comes with the first support for "curve" types, based on the International Organization for Standardization (ISO) SQL/MM (SQL Multimedia and Application Packages) model for curves. Also initial support for the ISO SQL/MM suite of spatial database functions is implemented [17].

In addition to the ongoing implementation of the ISO SQL/MM standard the PostGIS team works on three-dimensional surface and spline curve support, topology, networks, routing, long transactions and raster integration. The initial groundwork for using PostGIS as an ESRI ArcSDE style interface was also laid in version 1.2. This includes support for most of the ST_* and SE_* spatial SQL functions used by the ArcSDE spatial SQL interfaces.

Cross References

► Data Infrastructure, Spatial
► deegree Free Software
► Dimensionally Extended Nine-Intersection Model (DE-9IM)
► OGC's Open Standards for Geospatial Interoperability
► Oracle Spatial, Geometries
► University of Minnesota (UMN) Map Server

P

Recommended Reading

1. Free Software Foundation (1991) GNU General Public License. http://www.gnu.org/copyleft/gpl.html. Accessed 1 Jan 2007
2. GEOS (2006) Geometry Engine Open Source. http://geos.refractions.net. Accessed 1 Jan 2007
3. PROJ.4 (2000) Cartographic Projections Library. http://www.remotesensing.org/proj. Accessed 1 Jan 2007
4. Refractions Research (2007) PostGIS. http://postgis.refractions.net/. Accessed 1 Jan 2007
5. University of California (Berkeley) (1999) The BSD License. http://www.opensource.org/licenses/bsd-license.php. Accessed 1 Jan 2007
6. PostgreSQL Global Development Group (2007) PostgrSQL. http://www.postgresql.org/. Accessed 1 Jan 2007
7. Shekhar, S., Chawla, S.: Spatial Databases: A Tour. Prentice Hall, Upper Saddle River, NJ (2003)
8. Rigaux, P., Scholl, M., Voisard, A.: Spatial Databases: With Application to GIS. Morgan Kaufmann, San Fransisco (2002)
9. OGC (ed.): OpenGIS Simple Features Specification for SQL (Revision 1.1). OGC (1999)
10. ISO/TC211: ISO 19107 Geographic Information—Spatial Schema. ISO (2003)
11. OGP Surveying & Positioning Committee (2006) EPSG Geodetic Parameter Dataset v 6.11.2. http://www.epsg.org. Accessed 1 Jan 2007

12. The GiST Indexing Project (1999) GiST: A Generalized Search Tree for Secondary Storage. http://gist.cs.berkeley.edu. Accessed 1 Jan 2007
13. Sigaev, T., Bartunov, O. (2006) GiST for PostgreSQL. http://www.sai.msu.su/~megera/postgres/gist. Accessed 1 Jan 2007
14. Hellerstein, J.M., Naughton, J.F., Pfeffer, A.: Generalized search trees for database systems. In: Proceedings of the 21st International Conference on Very Large Data Bases, Zürich, Switzerland, 11–15 Sept 1995
15. Guttman, A.: R-trees: a dynamic index structure for spatial searching. In: Proceedings of ACM SIGMOD International Conference on Management of Data Boston, MA, 18–21 June 1984
16. Nebert, D.D. (2004) Developing Spatial Data Infrastructures: The SDI Cookbook, Version 2.0. http://www.gsdi.org/pubs/cookbook/cookbookV2.0.pdf. Accessed 12 Oct 2005
17. ISO/IEC: ISO/IEC 13249-3: Information Technology—Database Languages—SQL Multimedia and Application Packages. Part 3: Spatial (2006)

Postgres

► PostGIS

Preference Structure

► Multicriteria Decision Making, Spatial

Prism, Network Time

► Time Geography

Prism, Space-Time

► Time Geography

Privacy

► Cloaking Algorithms for Location Privacy

Privacy and Security Challenges in GIS

BHAVANI THURAISINGHAM, LATIFUR KHAN,
GANESH SUBBIAH, ASHRAFUL ALAM,
MURAT KANTARCIOGLU
Department of Computer Science,
The University of Texas at Dallas, Dallas, TX, USA

Synonyms

Geographic data management

Definition

Geospatial data refers to information about shapes and extent of geographic entities along with their locations on the surface of the earth. This definition, however, is often extended to include any physical or logical entity as long as it exhibits one or more geographic characteristics such as topology of a proposed highway infrastructure or location of a moving vehicle. Geospatial data management pertains to the acquisition, manipulation and dissemination of geospatial data under a set of guidelines. It has numerous applications including counter-terrorism, climate-change detection and space exploration. For example, global warming has been one of the major climate changing events in recent years. The significance of global warming lies in the severe impact that even small climate changes could cause on weather patterns, ecosystems and other activities. Understanding the causes and impacts of global warming is therefore critical. Central to this mission are the thousands of stations capturing vast amounts of geospatially referenced climate and weather data, both on and off the Earth. The data is stored in hundreds of geographically distributed databases, often in different formats. Even more problematic is that the data lack a common semantics, and as a result tends to take on different meanings in different places. These two problems are major impediments to scientists in their ability to coherently and consistently analyze the data, and investigate global trends, make predictions, and so forth.

One way to effectively analyze and detect climate changes is to apply knowledge discovery techniques, also referred to as data mining, for geospatial data sources. If the experts are to systematically process the data in order to answer important scientific and social questions, a coherent representation of the geospatial data related to global warming is needed. The semantic heterogeneity problem is handled by establishing domain ontologies (e. g., emission model, temperature model, sea-level model) to aid in the process of data annotation. A large number of existing environmental parameters can be mapped to geospatial data objects and the remaining ones could be added on gradually.

While the geospatial data related to climate modeling and changes, as well as much of the geospatial data such for counter-terrorism applications such as photographs of building and bridges, are usually publicly available, certain fields may be sensitive to a particular organization. Furthermore, the results of the integration and analysis of the geospatial data may also be sensitive. A recent report by Rand Corporation has stated that geospatial data, even those publicly available, have security needs that must be dealt with [1]. National Oceanic and Atmosheric Administration (NOAA) has also discussed the strong need for security policy enforcement for climate data records (CDR) [2].

Much progress has been made on geospatial information systems such as the specification of the geospatial markup language (GML) [3] for data representations by organizations such as the Open Geospatial Consortium (OGC) as well as information retrieval techniques. However several areas including techniques for integrating geospatial data as well as mining the data needs research. Furthermore, security and privacy issues have received very little attention for geospatial data management integration and mining. Research in the areas of geospatial data integration, mining and security are being conducted.

Historical Background

Some past research work has been reported on secure geospatial data management systems [4,5], as well as secure web services and secure semantic web [6]. For example, Atluri [4] has proposed a model that takes into account the characteristics of geospatial data. Bertino et al. [7] have developed a model called GEORBAC that extends role-based access control (RBAC) for geospatial data that take into consideration classification policies depending on content, content and time. The OGC members have also done some exploratory work in the use of Public Key Infrastructure (PKI) and extensible access control markup language (XACML) for building and deploying more secure geospatial portal applications. The OGC is also working on standards for geospatial digital rights management. However, in the literature survey done there is no work on developing secure geospatial semantic web and web services except for the research being conducted at the University of Texas at Dallas [8].

In a service-oriented architecture or a distributed system where multiple parties collaborate to exchange geospatial data, it is imperative that a strong security mechanism is maintained to ensure participating parties' continued willingness to share data. The abundance of data exchange protocols and the varying business needs of the parties make it a challenging task to devise an appropriate security model. The security specification from the Organization for the Advancement of Structured Information Standards (OASIS) defines a web service security model that unifies several popular security models and technologies to be able to interoperate in a platform- and language-neutral manner. XACML is the OASIS security standard, which allows developers to write and enforce information access policies for web services. The web service policy language (WSPL) is another proposed language for web services security framework. These languages lack inference and reasoning capabilities as they are not semantics-aware frameworks for machines to interpret, although they establish syntactical interoperability. GeoXACML [9] is an access control language proposed for geospatial web services.

There are two overlooked aspects in the existing security models mentioned above. First, they are mainly suitable for a single-party environment. In an integrated environment where resources come from various parties, the individual policies of each party have to be combined to apply in a global context. Bertino et al. [10] have proposed an integration algorithm for combining access policies of multiple autonomous parties in a distributed environment. They extend XACML by including a set of preferences that allow dynamic computation of policy integration need. The other overlooked aspect in the current models is the lack of semantics awareness in policy constructs. Semantic Web allows a platform for policy reasoning and inferencing if the policies are written in a semantic-aware language. Although the techniques for Semantic Web security are yet to be standardized, there has been work involving security ontologies. Different policy representations have been proposed using semantic languages such as Rei, and KAoS. KAoS exploits ontologies for representing and reasoning about domains describing organizations of humans and agents. Rei is a deontic concept-based policy language in Resource Description Framework-Schema (RDF-S).

One of the major challenges confronted in geospatial data management is collection and assimilation of data without major loss of fidelity. The most commonly employed approach has been using geospatial systems or ad-hoc programs to define methods that convert data from one source or format to another with the help of wrappers. This approach has limitations in so far as the wrappers are cumbersome and require manual translations every time a new data format or standard appears. Several proposals have been offered that utilize schema mechanisms (e. g., GML) to define concepts in a standardized manner. Nonetheless, the semantics provided by the schemas for geospatial resources are not machine-readable and hence are difficult to share between systems without prior coordination. While there have been researches to address these limitations (e. g., [11]), a comprehensive approach to developing a geospatial semantic web with appropriate technologies for specifying semantics as reasoning engines are yet to be developed.

Scientific Fundamentals

There are different levels of interoperability issues that need to be addressed when two or more geospatial data sources are to be integrated. One of the major problems is semantic heterogeneity. An example is the following: land cover classifications where definitions of forest, plantation, wood, copse, scrub, orchard, etc. all relate to areas with some tree cover but different organizations and countries may use them differently as well as use different terms for the same entity. The other problem is structural heterogeneity. For instance, a geographic location can be expressed by, for example, a closed string, and two separate coordinates or by a point. Research on semantic interoperability between geospatial data sources of the same theme is underway. A major challenge is to integrate the work of OGC and the World Wide Web Consortium (W3C) to develop a geospatial semantic web that handles semantic heterogeneity. Another challenge is in the development of geospatial semantic web services that can discover and manage resources in a global environment.

While integration of data sources is important, it has to be done securely to ensure participating parties' willingness in sharing their data. An important security consideration in this process is the integration of security policies. Since the individual agencies implement their own security policies to protect the data, several critical issues arise during the policy integration. The first issue is the mismatch of policy rule semantics. That is, when a policy has to be integrated with other policies, attributes and targets of the policies should be interpreted consistently by the system. For example, if two policies from separate agencies use "manager" and "supervisor" respectively, to specify the same role attribute, the integration algorithm should be able to interpret this equivalency. The second issue is rules mismatch. Even if the assumption of no heterogeneity is made, attributes sets and targets of separate policies have to be matched properly.

Further security challenges include coming up with appropriate policies for climate and weather data, as well as language to specify the policies. Policies may depend on content, context and time. Different agencies may enforce different policies. Furthermore, collections of data from multiple databases within an agency or from multiple agencies taken together may be sensitive, while individually they may not be classified. The geospatial semantic web is expected to provide a level of semantics to help in designing secure contextualized and georeferenced policies that reason about their robustness.

A study was conducted to evaluate existing geospatial web service standards against the requirements identified in the use cases, in particular, identification of formal change requests to enhance existing standards. In those cases where existing standards will not work or cannot be adapted, identifying and developing new web service interface standards was investigated. In both cases, the focus was on (1) geospatial semantic web services for applications such as discovering and managing geospatial data resources, and (2) geospatial semantic web technologies for information integration and related security considerations. Both types of services are closely intertwined as the information integration application will invoke the geospatial semantic web services for providing various services. Each web service has a high-level service description that is written using Web Ontology Language for Services (OWL-S). OGC specifies geospatial interface and encoding standards. The key encoding standard is the GML. OWL-S provides a semantic rich application level platform to encode the web service metadata using descriptive logic. The approach used is essentially the following:

- Semantic enrichment of the OGC web services framework by using OWL-S ontology
- Query disambiguation of the service requestor using semantics
- Automatic service discovery and selection using capability-based matchmaking
- Automatic service composition and invocation

Since geospatial data involves geospatial constructs such as overlap and boundary which are required to be disambiguated during the query phase, the registered services will then be automatically discovered for the disambiguated query using capability-based search which is a more expressive mechanism than the simple keyword-based search currently used in the service registries. The selected web services will be automatically invoked using the WSDL groundings. Dynamic service compositions on the fly are made possible for the service requestor's query. The research carried is for developing geospatial semantic web technologies for information integration. Development of a geospatial resource description framework (GRDF) that extends GML to include semantics [8] has been initiated. This is also intendend to enhance GRDF (e. g., extensions to support climate data), which is the foundation for a geospatial semantic web, and subsequently extend the reasoning engines (such as those in JENA; JENA is a java framework for building semantic web applications) for geospatial data.

Research is also needed to investigate security issues for geospatial semantic web and web services. The core of the approach is represented by a semantically rich web service access control model consisting of a policy layer that processes user queries to geospatial web service agents. The security policies have to be enforced and only the authorized data is retrieved and returned to the user. In the case of multiple geospatial data servers, each node may enforce its own set of policies as specified and enforced by the policy framework. Data access by a web service is mediated by a broker and the request is then sent to different locations. Since policy descriptions and granularity will be annotated in descriptive logic (i. e., OWL-DL), the proposed access control model will allow automatic reasoning between communicating clients and agents. A secure GRDF language is being developed examined to specify the security semantics.

There are unique challenges for discovering knowledge from climate-change-specific geospatial data. For example, Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) data is used to model detailed maps of land surface temperature, emissivity, reflectivity, and elevation. This characteristic of ASTER data offers opportunities to observe, understand, and model the Earth system, enabling us to better predict change, and to understand the consequences for life on Earth. ASTER obtains high-resolution (15–90 m^2 pixel) images of the Earth in 14 different wavelengths of the electromagnetic spectrum, ranging from visible to thermal infrared light. Therefore, there is a need to build higher-level knowledge from this data for analyzing complex phenomena. Geospatial data specific to climate change has spatial and temporal characteristics that add substantial complexity to data-mining tasks. The spatial relations, both metric (such as distance) and nonmetric (such as topology, direction, shape, etc.) and the temporal relations (such as before and after) are information bearing and therefore need to be considered in the data mining methods.

Data mining raises serious security and privacy concerns. There are two aspects here; one is that the results of data mining may be sensitive. The other is that while the individual climate data records are sensitive, the results of the data mining tool are unclassified. These issues have to be investigated for geospatial data.

For climate change, current work focuses mainly on the change detection of various classes (i. e., "urban area", "forest" and so on) that appear in images of a particular location over time. The tasks involved for such an approach include identifying the class/label of pixels in images, estimating contiguous areas in the map/image that belong to the same class, and comparing areas of the same class taken from two different images for the same location and determining changes. For example, from 1986 to 1998, urban areas increased a total of 52,019 ha or by 28.4%. This number can be estimated by first classifying urban areas in images for 1986 and 1998 separately and then estimating the difference. To classify pixel value into various classes, the current state-of-the art uses a maximum likelihood (ML) classifier; it has been observed that the accuracy of ML is not satisfactory. Lower accuracy may contribute higher false positives and higher false negatives for climate change detection [12].

As far as the authors know, security for geospatial data mining has not received any attention. At the University of Texas at Dallas research has started in this field with respect to both confidentiality and privacy.

The approach consists of the following:

- Extracting features to facilitate climate change detection
- Training classifiers using extracted features and predicting class/label of pixels that appear in images
- Comparing contiguous areas of the same class taken from two different images for the same location to facilitate change detection
- Correlating these atomic concepts/classes to make a decision of generic concept with the help of ontologies

For feature extraction, ASTER data has 14 channels, from visible through the thermal infrared regions of the electromagnetic spectrum, providing detailed information on surface temperature, emissivity, reflectance, and elevation. ASTER provides valuable scientific and practical data of the Earth in various fields of research. To classify pixels that appear in images, research is by exploiting various data mining techniques including support vector machines (SVM) combined with a developed technique called Dynamically Growing Self Organizing Tree (DGSOT) [13]. Investigation has shown that SVM+DGSOT is a powerful method for classification. This classifier will help to determine atomic classes/concepts. Change detection can be done by comparing contiguous areas of the same class taken from two different images for the same location. Exploiting ontologies with embedded rules will enable the determination of generic concept/outcome. For example, a set of high-level concepts (i. e., wildfire) can be inferred using ontologies and a set of atomic concepts (e. g., low rainfall). In particular, exploiting ontology-based concept learning improves the accuracy of the individual concept. This is achieved by considering the possible influence relations between concepts based on the given ontology hierarchy.

Two aspects with respect to security need examination. First, the prior research on enforcing security and privacy constraints for data management systems must be examined, and the inferencing techniques for classifying the results produced by the data mining tools applied. Previous work in secure multiparty-based cryptographic approaches for privacy preserving data mining as well as other approaches should also be examined, and techniques developed for security/privacy preserving geospatial data mining [14].

Key Applications

Geospatial data are becoming increasingly useful across many different applications for enhancing the visual aspect

P

of the raw data and providing additional dimensions to enable decision making and analysis. Some of the most promising and critical applications are described here.

Emergency Response System

In the case of an emergency, first responders and decision-making personnel often need to gather and analyze georeferenced data on the fly. Without efficient data management, collecting and presenting the pertinent data in a coherent form would be unfeasible.

Climatology

Geospatial data includes information regarding weather patterns, seasonal changes, wind velocity, and atmospheric and sea-level pressure and so on. Proper collection and filtering of this data is critical in studying climate trends. Climate changes that are deviating from the norm or that imply serious repercussions can be determined based on the collected data.

Semantic Web

Semantic web refers to a distributed system where all kinds of data stores and client applications are connected via a framework that incorporates a loose data model, logic, rules and reasoning. The basic idea behind semantic web is to enable a minimum human-intervention infrastructure and maximum machine automation. The applications on the semantic web can tap into various data sources to fetch the pertinent data, and then merge them to present coherent and precise results to application users. For instance, a semantic-web-enabled automated restaurant finder agent can extract restaurant data and georeferenced data to present not only the route to the destination, but the weather and crime rate in the area as well.

Future Directions

This paper has provided an overview of geospatial data management and discussed the need for security;, geospatial data integration, geospatial data mining and the impact of security and privacy on these functions have been discussed. For each of the functions, challenges have been identified, along with the state of the art and research directions.

As stated earlier, security and privacy are important considerations for geospatial data mining. Even through much of the geospatial data is publicly available, according to the Rand report there are many attributes that have to be protected. Furthermore, the privacy of the individuals has to be maintained. There is still much work to be done in geospatial data interaction, mining, security and privacy.

Acknowledgements

Special thanks go to Dr. Mike Jackson of the University of Nottingham and the Open Geospatial Consortium for their comments on our research. Prof. Elisa Bertino and Prof. Michael Gertz also provided valuable comments on geospatial data security.

Recommended Reading

1. Assessing the Homeland Security Implications of Publicly Available Geospatial Information. Rand Report for NGA (2004)
2. National Research Council: Climate Data Records from Environmental Satellites: Interim Report, NOAA Operational Satellites. National Research Council (2004)
3. Geography Markup Language (GML) Version 3.1.1. http://portal.opengeospatial.org/files/?artifact_id=4700
4. Atluri, V., Chun, S.: An authorization model for geospatial data. IEEE Trans. Dependable Sec. Comput. **1**, 238–254 (2004)
5. Geospatial Interoperability Reference Model (GIRM, V 1.1) http://gai.fgdc.gov/. Accessed December 2003
6. Lieberman, J., Pehle, T., Dean, M.: Semantic Evolution of Geospatial Web Services. http://www.w3.org/2005/04/FSWS/Submissions/48/GSWS_Position_Paper.html
7. Belussi, A., Bertino, E., Catania, B., Damiani, M.L., Nucita, A: An authorization model for geographical maps. http://www.informatik.uni-trier.de/~ley/db/conf/gis/gis2004.html GIS 2004
8. Ashraful, A., Thuraisingham, B.: Geospatial resource description framework (GRDF) and secure GRDF. Tech. Rep. UTDCS-03–06, University of Texas at Dallas (2006) http://www.cs.utdallas.edu
9. Matheus, A.: Declaration and enforcement of fine-grained access restrictions for a service-based geospatial data infrastructure. In: Proceedings of the10th ACM Symposium on Access Control Models and Technologies, pp. 21–28. ACM, New York (2005)
10. Mazzoleni, P., Bertino, E., Crispo, B., Sivasubramanian, S.:XACML policy integration algorithms:~not to be confused with XACML policy combination algorithms! In: Proceedings of 11th ACM Symposium on Access Control Models and Technologies, pp. 219–227. ACM, New York (2006)
11. Li, D.: Geospatial Semantic Web Research at LAITS. http://www.ncgia.ucsb.edu/projects/nga/docs/Di_Position.pdf
12. Buttenfield, B., Gahegan, M., Miller, H. Geospatial Data Mining and Knowledge Discovery. http://www.ucgis.org/priorities/research/research_white/2000%20Papers/emerging/gkd.pdf
13. Khan, L., Awad, M., Thuraisingham, B.: A new intrusion detection system using support vector machines and hierarchical clustering. VLDB J. (2007, in press)
14. Kantarcioglu, M., Clifton, C.: Privacy-preserving distributed mining of association rules on horizontally partitioned data. IEEE Trans. Knowl. Data Eng. **16**, 1026–1037 (2004)
15. Onsrud, H.J., Johnson J.P., Lopez, X.: Protecting personal privacy in using geographic information systems. Photogramm. Eng. Image Process. **60**:1083–1095 (1994)

Privacy of Location Traces

► Privacy Preservation of GPS Traces

Privacy Preservation of GPS Traces

MARCO GRUTESER, BAIK HOH
Electrical + Computer Engineering (WINLAB),
Rutgers University, Piscataway, NJ, USA

Synonyms

Privacy of location traces; Anonymization of GPS traces; Privacy preserving in location based services

Definition

Techniques to evaluate and enhance the privacy of users that contribute traces of their movements to a geographical information system (GIS). Due to the increasing prevalence of global positioning system (GPS) chips in consumer electronics and advances in wireless networking, GIS can collect *GPS traces* of large numbers of individual users. These traces give rise to privacy concerns, since GPS traces can reveal visits to sensitive or private places (e. g., home, medical clinics) and associated information such as time of day or speed of travel.

Privacy can be enhanced through standard data protection techniques such as policy disclosure, obtaining user consent, access control, and encryption. Anonymization, another standard technique, is of particular interest for GIS applications that aggregate data from many users, since it also protects against accidental and insider data breaches and enables public release of aggregate datasets. Anonymization of location traces, however, poses special challenges since the detailed time-series nature of a GPS trace often allows re-identification of users. For example, it is often straightforward to identify the home or work positions based on a GPS trace, providing means to re-identify the user. Thus, removing identity information from a trace only provides weak anonymity. To obtain a strong degree of anonymity, special location disclosure control algorithms must be used that reduce resolution, omit especially sensitive parts of the trace and divide traces into shorter disjoint parts to prevent extended tracking.

Historical Background

Privacy concerns due to the misuse of new technological inventions can be traced back at least to Louis Brandeis article "The right to privacy" [4] addressing photography in 1890. Since then technological advances have posed repeated challenges and have given rise to new social norms, legal concepts, and technological solutions. Early technological solutions for data privacy include data encryption for communication and storage, operating system and database access control, and auditing. Over the past few decades, as information technology has permeated our lives, several notable privacy technology developments occurred that have influenced the development on techniques for GPS traces:

- *Statistical Databases.* These databases allow queries that retrieve aggregate statistics about individuals' data but do not allow retrieval of any individual's record. The main challenge in the development of such databases is protection against inferences that reconstruct individual records from the results of several carefully selected aggregate queries.
- *Privacy-Aware Data Mining.* The objective of these techniques is similar to statistical databases in that they seek to protect individual database records. Instead of calculating precise statistics, these algorithms only allow reconstruction of the approximate distribution of attributes over the total population, which is sufficient for many data mining tasks. For example, privacy techniques, such as value-class membership or time-series distortion, can increase privacy for individual records, while still allowing classifications algorithms to operate on aggregate data [1].
- *k-Anonymity.* The k-anonymity concept provides a formal model for evaluating privacy protection of a dataset. Samarati and Sweeney [14] developed this concept and an algorithm that can remove or generalize sensitive data so that a user's record is indistinguishable from at least $k-1$ other records. Thus, this algorithm enables anonymizing a database table, so that the table can subsequently be released to external sources (e. g., releasing medical records to researchers). Anonymity-based solutions were also developed for enhancing communication privacy [5].

These concepts provide a foundation for the development of location privacy techniques described in the following sections, which were motivated by the advent of affordable positioning and tracking technologies.

Scientific Fundamentals

A typical GPS trace contains a collection of individual position samples, each comprising latitude, longitude, timestamp, and optionally speed and heading. Privacy of a dataset of such traces may be protected through well-known data protection techniques, such as encryption and access control. These techniques are effective, when only a limited number of fully trustworthy users require access to the dataset. The dataset can then be protected from eavesdroppers or curious other users by encrypting the dataset before communication or storage, for example.

Anonymization techniques may be more appropriate if the dataset must be released to a larger number of not fully trusted parties or when the identities of data providers are not needed for the application. The remainder of this entry will concentrate on such techniques, since the techniques used in this case a more specific to GPS Traces.

A first step towards effective anonymization is removing explicit user identifiers, such as names or cell phone identifiers, that may be associated with the trace. We refer to this as a (weak) anonymous trace.

The exact privacy implications of such anonymous GPS traces depend on many factors, especially GPS accuracy, building density, sampling frequency, trace duration, user density, and other data associated with each sample. First, consider a trivial anonymous GPS trace containing only a single position (latitude, longitude, timestamp). This trace could pose a location privacy risk, if an adversary can infer the user's identity. Identification is possible through

- *Restricted space correlation.* A restricted space is a geographic area only accessible to one known person, such as a home or office. If samples originate from this location, the adversary can infer with high probability the user identity associated with this sample.
- *External observation correlation.* An external observation is a sighting of a single known individual at a give position and time through other means (e. g., electronic toll booth records, credit card transaction data, video surveillance tape, etc.). If no other uses were present at the location and the position and time of the sighting matches the GPS sample, the user can be identified.

Both methods require that some information about the position of the individual is already known. Still, privacy risks can exist when disclosing the GPS traces for three reasons. First, an adversary may learn more precise information about the whereabouts of the individual, for example the exact time an individual was present or the exact room a person visited. Second, the usage of GPS samples poses a more general *data privacy* risk, if other sensitive information is associated with the sample. For example, a user might conduct an apparently anonymous location-based search for the closest medical clinics on a cell phone, which sends a GPS sample associated with the search terms to an external search service provider. If one of the above identification methods is possible, an adversary with access to the service provider logs may connect the search terms (e. g., a medical condition) with a particular individual by using the GPS sample. Third, the traces may reveal information about other visits and activities, if they contain multiple samples. We will discuss this case further below.

Generally, identification through unrestricted space and external observation correlation is feasible, if correlation data is available and the GPS resolution is high enough to uniquely identify a person or space from the correlation data. Modern GPS receivers typically achieve sub 10m accuracy in open-sky areas, enough to uniquely identify most suburban homes, but rarely sufficient to pinpoint an apartment in an urban high-rise building. A similar relationship between GPS accuracy and user density exists for observation identification.

Spatial cloaking [8] provides a countermeasure against these risks. It dynamically adjusts the resolution of position samples to maintain a constant degree of privacy in situations with different user densities. Given a set of traces from different users, the spatial cloaking algorithm achieves k-anonymity by determining a square that encloses the current positions of at least k users. Square corners are chosen from an external reference grid, so that they do not reveal any clues about current user positions. The position samples of the k users are then replaced with the square (or its center point).

The privacy risks for single positions are compounded for longer GPS traces, which contain more than one position sample. If a user can be identified at any one point, an adversary can infer which buildings (e. g., stores, clubs, medical clinics, entertainment venues) a person visited and accurately measure time spend at work or at home. If the frequency of location samples is high (at least one every few minutes) one may also infer speed limit violations while driving, for example, even if the GPS device does not report speed information. Further identification risks are higher, because a person could now be identified through knowledge about the frequency of their visits to each location in the trace [11].

A countermeasure against these particular trace risks is *path segmentation* [2,9,15], which divides several anonymous traces into shorter traces, or in the extreme, into a set of anonymous samples. Intuitively, this might reduce the risks to those identified for anonymous samples. However, an adversary may frequently be able to reconstruct the complete traces by "following the footsteps" (if one segment begins where another one ends the trajectory of both points into the same direction, they likely belong together). This can be automated through location tracking algorithms that exploit the spatio-temporal correlation between subsequent samples, such as *multiple target tracking* [9,13]. In essence, these algorithms predict a user's next position based on the previous trajectory and add the sample closest to the prediction to the trace. This approach fails, if many potential users are near the predicted position—thus, the segmentation approach is only effective in areas where user density is high and many users share common paths. Note that the target tracking algorithms can also filter noise from the location samples, thus privacy techniques that add random noise to each sample may be ineffective, unless the noise component is very large compared to the range of possible positions.

Better privacy protection for GPS traces can also be provided through special disclosure control algorithms such as *origin-destination cloaking* (ODC) [10] or *uncertainty-aware path cloaking* [12]. ODC is designed for GIS applications that primarily require applications from moving users, such as traffic monitoring applications in the automotive domain. ODC cloaking aims to suppresses the parts of location traces that are close to locations that a user has visited, but allow release of location information when the user is moving. The intuition behind this approach is that visited locations provide likely avenues for identification and reveal potentially sensitive information. With ODC the exact visited building remains hidden, only the general area is known. Thus, both restricted space identification and compiling a dossier of visited locations becomes more difficult. Uncertainty-aware path cloaking further limits the tracking time when moving by dropping samples when extended tracking was possible.

Key Applications

Many pervasive or context-aware computing applications rely on the availability of periodic and accurate location information provided by ever more cost-effective GPS chips. Applications such as the following that make GPS traces available to external service providers can benefit from the described data privacy techniques:

- Traffic monitoring applications: Instead of camera or loop detectors on the roads, probe vehicles, which are equipped with GPS and sensors, are expected to be used in many traffic monitoring systems [11]. Usage of the described privacy techniques could better protect privacy and increase participation rates in such schemes.
- “Pay as you drive” insurance: This approach allows auto insurance carriers to customize insurance premiums to individual driving patterns. In return for potential discounts, drivers let the insurer install a GPS device that provides GPS traces to the insurer. To improve risk assessment the insurer can then analyze the traces for mileage driven, roads taken, speed, time of day for trips, duration of rest periods, and other factors. While this application likely requires drivers to identify themselves to the insurance provider, techniques like OD cloaking may also be beneficial in this scenario, to reduce the amount of information collected.
- Electronic toll payment: Some next-generation electronic toll collection systems use GPS to calculate more fine-grained distance and time-based tolls. Current radio-tag based systems (e. g., EZ-Pass in New Jersey and FasTrak in California) have been regarded with suspicion since the history of road usages are collected with identity, this lets authority clearly see where subscribers are driving. However, recent research [3,6] proposed more privacy-aware toll collection protocols.
- Cell phone location-based services: Many US cell phone handsets incorporate GPS chips that can provide very precise position information in many cases. These are used primarily to satisfy the E911 regulatory requirements, which mandate that cell phone service providers must be able to locate emergency callers. This infrastructure, however, is also being used for offering location-based services, such as point-of-interest queries or navigation. Spatial cloaking allows users to use these services with enhanced privacy.

Future Directions

Applications that have access to private GPS traces from large numbers of users are relatively new. Thus this area provides many topics for further research.

- *Risk analysis and privacy metrics.* To date little practical experience with such applications exist. Privacy risks are typically identified by studying analogies to risks in other information systems. Improved privacy frameworks and metrics are needed to guide analysis of privacy risks in applications. These frameworks should include quantitative guidance on parameters such as user density, sampling frequency and trace duration.
- *Usable privacy preferences* Since increased privacy protection usually reduces the quality of service provided by the application, a complete privacy solution should allow users to choose or specify different disclosure options. This requires research on user interfaces to understand how users can best express these preferences. It also requires research in privacy algorithms that must remain secure even if some users disclose more detailed information than others.
- *Maintaining privacy when using multiple techniques.* When different anonymization techniques are simultaneously used, for example to satisfy different application requirements, an adversary with access to the different produced datasets may be able to infer private information. Further work is needed in understanding these risks and offering appropriate solutions.
- *Analysis and penetration testing*. The described privacy algorithms are relatively new and should be subjected to more rigorous security analysis. As with other security techniques, only continued analysis and penetration testing over time will provide a good understanding of the exact level of protection they offer.

Cross References

► Multiple Target Tracking

P

Recommended Reading

1. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: Proc. of the ACM SIGMOD Conference on Management of Data, May 2000, pp. 439–450. ACM Press, Dallas, Texas (2000)
2. Beresford, A., Stajano, F.: Mix zones: User privacy in location-aware services. In: IEEE Workshop on Pervasive Computing and Communication Security (PerSec), pp. 127–131. IEEE Computer Society, Washington, DC, USA (2004)
3. Blumberg, A.J., Keeler, L., Shelat, A.: Automated traffic enforcement which respects driver privacy. In: Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems, Vienna, Austria, September 2005, pp. 650–655. IEEE Computer Society, Washington, DC, USA (2005)
4. Brandeis, L., Warren, S.: The right to privacy. Harvard Law Rev. **4**(5), 193–220 (1890)
5. Chaum, D.: Security without identification: Transaction systems to make big brother obsolete. Commun. ACM **28**(10), 1030–1044 (1985)
6. Choi, J.Y., Jakobsson, M., Wetzel, S.: Balancing auditability and privacy in vehicular networks. In: Q2SWinet '05: Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks, Montreal, Quebec, Canada, 2005, pp. 79–87. ACM Press, New York, NY, USA (2005)
7. Civilis, A., Pakalnis, S.: Techniques for efficient road-network-based tracking of moving objects. IEEE Trans. Knowl. Data Eng. **17**(5), 698–712 (2005) Senior Member-Christian S. Jensen
8. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceedings of the First International Conference on Mobile Systems, Applications, and Services, San Francisco, CA, pp. 31–42. ACM Press, New York, NY, USA (2003)
9. Hoh, B., Gruteser, M.: Protecting location privacy through path confusion. In: Proceedings of IEEE/Create-Net SecureComm 2005, Athens, Greece, September 2005, pp. 194–205. IEEE Computer Society, Washington, DC, USA (2005)
10. Hoh, B., Gruteser, M., Xiong, H., Alrabady, A.: Enhancing privacy preservation of anonymous location sampling techniques in traffic monitoring systems. In: Proceedings (Poster Session) of IEEE/Create-Net SecureComm 2006, Baltimore, Maryland August 2006, pp. 1–3. IEEE Computer Society, Washington, DC, USA (2006)
11. Hoh, B., Gruteser, M., Xiong, H., Alrabady, A.: Enhancing security and privacy in traffic-monitoring systems. IEEE Pervasive Comput. **5**(4), 38–46 (2006)
12. Hoh, B., Gruteser, M., Xiong, H., Alrabady, A.: Preserving Privacy in GPS Traces via Uncertainty-Aware Path Cloaking. In: CCS '07: Proceedings of the 14th ACM conference on Computer and communications security, Alexandria, Virginia, USA. ACM Press, New York, NY, USA (2007, in press)
13. Reid, D.: An algorithm for tracking multiple targets. IEEE Trans. Autom. Control **24**(6), 843–854 (1979)
14. Samarati, P., Sweeney, L.: Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression (Technical Report SRI-CSL-98-04). Computer Science Laboratory, SRI International (1998)
15. Li, M., Sampigethaya, K., Huang, L., Poovendran, R.: Swing & swap: user-centric approaches towards maximizing location privacy. In: WPES '06: Proceedings of the 5th ACM workshop on Privacy in electronic society, Alexandria, Virginia, USA, pp. 19–28. ACM Press, New York, NY, USA (2006). doi: 10.1145/1179601.1179605

Privacy Preserving in Location Based Services

► Privacy Preservation of GPS Traces

Privacy Threats in Location-Based Services

CLAUDIO BETTINI[1], SERGIO MASCETTI[1], X. SEAN WANG[2]

[1] Department of Informatics and Communication (DICo), University of Milan, Milan, Italy

[2] Department of Computer Science, University of Vermont, Burlington, VT, USA

Synonyms

Anonymity in location-based services; Location trusted server; Location server; Geopriv group, IETF; Identity aware LBS; Identity unaware LBS; Access control; Gnerelization; K-anonymity

Definition

Location-based services (LBS) are those services that, based on the user's current position, can provide location-aware information. Typical examples are map and navigation services, services that provide information on close-by public resources (e. g., gas stations, bus stops, pharmacies, and ATM machines), services that provide localized news (e. g., weather forecasts, road constructions, etc.), emergency services (911, 118, etc.) as well as more personalized services like proximity marketing or friend-finder.

Private information refers to the information a user does not wish to be released associated with her identity. This includes political or religious orientation, health information, financial assets, or closeness to specific individuals or organizations. LBS services play a role in this context because both identity and private information can be directly or indirectly released through a single or a sequence of LBS requests. LBS requests can reveal, for example, a) information on the specific location of individuals at specific times, b) movement patterns (specific routes at specific times and their frequency), c) requests for sensitive services (closest temple for a specific religious worship), or d) personal points of interest (frequent visits to specific shops, clubs, or institutions).

A privacy threat occurs whenever the information contained in one or more requests issued by a given user can be used, possibly associated with external information, to associate the user identity with the private information. The study of privacy threats and protection techniques in LBS is a subtle and challenging research topic.

Historical Background

Most of the approaches proposed in the literature to protect LBS privacy consider scenarios that can be easily mapped to the one depicted in Fig. 1.

Three entities are involved in the scenario:

- The **User** invokes or subscribes to location-based remote services that are going to be provided to her mobile device.
- The **Location Trusted Server (LTS)** is supposed to have access to the precise location data of a large group of users, act as a proxy for all LBS requests from these users and to enforce privacy policies for those requests.
- The **Service Provider (SP)** fulfills user requests and communicates with the user through the LTS.

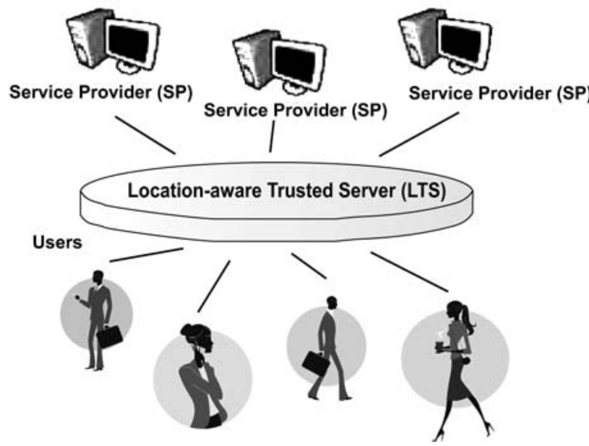
As pointed out above, this can be considered the reference scenario for many of the existing approaches to the privacy problem. Considering the model proposed by the IETF geopriv working group [6], the so-called *Rule Holder* and *Location Server* can be mapped as a single entity to the Location Trusted Server in the reference scenario; the *Location Generator* and *Location Recipient* can be mapped to the User device/infrastructure and Service Provider, respectively. The existence of an intermediate entity between the user and the service provider, possibly organized in multiple modules, is assumed also in [4,7,10,11,13,14,16].

Each user's request contains parameters concerning the identity of the user, the current time and position, and the requested service. While the specific request message may have an articulated structure depending on the specific parameters, three logical components can be identified:

$$r = \langle IDdata, STdata, SSdata \rangle$$

IDdata contains information on the user identity, *STdata* contains the requested spatio-temporal information, and *SSdata* specifies the service provider and the specific service parameters.

Any request in the above logical format may be the potential source used by attackers to violate user privacy. Requests, once forwarded by the LTS, may be acquired by potential attackers in different ways: they may be stolen from SP storage, voluntarily published by the trusted parties, or they may be acquired by eavesdropping on the communication lines. On the contrary, the communication between the user and the LTS is considered as trusted and the data stored at the LTS is not considered accessible by the attacker. Any entity that has access to the data of the SP or that can intercept the data in the communication channel between the LTS and the SP is a potential attacker. Hence, without loss of generality, in the following sections, the SP is considered as the potential attacker.



Privacy Threats in Location-Based Services, Figure 1 The reference scenario

Scientific Fundamentals

A subtle problem in LBS privacy is the identification of the possible threats. Section A provides a categorization of the possible privacy threats; first considering isolated requests, then sequences of requests issued by the same user. In Section B, the various techniques proposed in the literature are reported and their goals are specified with respect to the identified privacy threat.

A Privacy Threats in LBS

Considering the literature that has appeared on this topic, this section characterizes the typical scenarios of LBS requests and the possible privacy threats. In the following, there is a distinction between services that need to be aware of the user's identity (*identity aware LBS*) and services that can work with a pseudo-id or in an anonymous way (*identity unaware LBS*).

A.1 Privacy Threats in a Static Scenario The specific threats involved in the submission of each LBS request are considered here.

s.1 Sensitive association: IDdata and STdata. When a user submits an LBS request, an attacker could associate the user's identity with her spatio-temporal

P

location. There are some situations in which this association is considered sensitive by the user. Examples include all the spatio-temporal locations that can reveal a user's private information such as health status, political affiliations or religious beliefs.

s.1a Identity aware LBS In this case an attacker can always associate the identity of the user with her spatio-temporal location.
Consider a service for localized news that collects user identities in order to charge the cost of the service. If a user submits the service request while attending a political demonstration, an attacker could associate user identity with her political affiliation.

s.1b Identity unaware LBS Though user identity is not explicitly transmitted to the SP, an attacker can infer it from SSdata and STdata. The problem of discovering the identity of a user from a combination of values that refers to her has been extensively investigated in the area of database systems [12,15].
Consider an LBS that provides driving directions based on current traffic conditions. The service is provided by a business for its employees usually working outside the business establishment. The service is anonymous since the business cannot monitor the movements of its employees due to legal restrictions. However, if a user performs a request from his suburban house, then an attacker can compute the address from the STdata and associate it with the identity. This information can be used by the business to understand that the employee is not where she is supposed to be at that moment.

s.2 Sensitive association: IDdata and SSdata When a user submits an LBS request, an attacker could associate the user's identity with the specific service and service parameters invoked by the request. This data may be itself private information, or may be used to obtain private information.

s.2a Identity aware case This is the case in which the identity of the issuer is explicitly contained in the request. This case does not involve handling of spatio-temporal information and therefore is not peculiar to LBS related privacy threats.

s.2b Identity unaware case As in s.1b, the attacker is not aware of the identity of the user but can infer it from SSdata or STdata. Differently from s.1b, in this case a user's private information is derived from SSdata.
Consider a location based friend-finder service that is designed to introduce a user to the people that are close by and that have similar interests. Anonymous users send requests providing, as SSdata, some personal data (gender, age, first name) and an interests profile (sport, music, etc.). Suppose the user sends a request from her apartment. From the STdata, the attacker can discover the address from which the request is sent. Then, from a voter list (or other publicly available source), the attacker can acquire a list of people living at that address and obtain some information about them, such as age, gender and first name. These values can be compared with the ones provided by the user to possibly discover an identity that is eventually associated with the user's interests.

Note that a special case is that the STdata and/or SSdata fields are empty. If both fields are empty, it could still be a source for privacy violation, e. g., in threat s.1a, in which the privacy information could simply be the fact that the user has made a request. This also applies to the threats in the dynamic scenarios detailed in the next subsection. Note, however, that such empty-field requests can be seen as having special implicit (or default) values for the requests themselves. For example, a request without any SSdata to a toll booth could implicitly contain SSdata "paying for toll". Empty STdata could implicitly mean that the location and time are irrelevant to the request and therefore can be taken similarly as "any possible location and any possible time".

A.2 Privacy Threats in a Dynamic Scenario It is very common that a user submits several requests to the same LBS. If the requests are not related, each of them can be considered individually as in the static scenario. On the other hand, if it is possible for an attacker to link the requests to the same user, new privacy threats are possible. Several techniques exist to link different requests to the same user. The most intuitive one is the observation of the same identity or pseudo-id. Since the ability of linking requests is not a threat in itself, these specific techniques are not herein addressed.

A *request trace* is a set of requests that the attacker can correctly associate to a single user. Analogously, a *STdata trace* (and *SSdata trace*) is the set of STdata (and SSdata) that is contained in a request trace.

In a dynamic context, a user can consider as sensitive the association of her identity with a trace of STdata or SSdata. In this case, the LTS has to guarantee that the user identity is not disclosed in the same request trace.

d.1 Sensitive association: IDdata with STdata or SSdata. Note that these are the same associations as in s.1 and s.2; however, in a dynamic scenario new threats are possible that reveal these associations.

There are situations in which a location, such as an office or a house, may identify a small group of people. However, a single request from that location may not be sufficient to identify the sender. For example a user can submit an LBS request from a colleague's office or from a house where she is not living in. However, if a user submits LBS requests several times from that location, an attacker can trace the requests and eventually identify the user.

For instance, an attacker can observe a user's movements that repeat frequently. Using empirical assumption about users habits (like "users go from home to work in the morning and back in the afternoon") some information can be acquired about the user.

Consider a service that provides driving directions based on current traffic conditions. A professor uses the service every day to go from home to the university and back. The attacker sees that the pattern going from the location of the house to the location of university and back is frequent (almost every working day) hence it can infer that the two locations correspond to house and work. A cross check of the list of university faculty and the list of people living in the house leads the attacker to obtain the identity of the professor.

d.2 Sensitive association: IDdata and STdata trace. There are situations in which STdata of a single request can reveal personal information (e. g., threat s.1). However, in general, a trace of STdata or SSdata can provide the attacker with more reliable information.

An attacker that observes through a STdata trace that a user goes to a Church every Sunday can deduce the user's religious beliefs with good reliability.

d.3 Sensitive association: IDdata and SSdata trace. In many cases, the private information that an attacker can obtain from SSdata trace is equivalent to the union of the pieces of information that the attacker would obtain from each single SSdata. However, in general, there is some private information that can only be inferred from a trace of SSdata and not from a single one.

Consider a disease that obliges a user to frequently call a medical emergency service. In each request the user specifies the symptoms. While a single request could not reveal the disease, if the attacker observes the SSdata trace, he could discover it.

d.4 Disclosure of visited locations. If an attacker knows a user's STdata trace, it could infer other positions the user visited even if no request was performed from those locations.

Consider a service for car accident monitoring. Each user is identified by the SP with a pseudo-id. Suppose that a user is traveling on a highway and frequently communicates her position to the SP. On the highway, there are cameras that recognize car plates (e. g., to charge the road toll). The user is aware that revealing her car plate number together with the service pseudo-id can lead an attacker to associate her identity with the trace (the user suspects that the attacker can also access the data from the cameras). Hence, the user temporarily suspends the service. However, the attacker knows the user's locations before and after the cameras and therefore can infer that the user traveled through the area where cameras are positioned. Considering the average speed, the attacker can also estimate when the user was there and hence can associate the identity with the trace.

B Privacy Preserving Techniques

In this section the different techniques that have been proposed to address the privacy threats are briefly illustrated.

B.1 Access Control Advanced access control models can be used in the context of LBS services to specify and enforce privacy policy rules. The rules can define, for example, the type of data that each service provider can access, the resolution of that data, and possibly other constraints. With respect to our reference model, policies can be defined by users as well as by service providers and can be enforced by the LTS. Among the efforts in this direction, in [16] a push-based LBS scenario is considered; users can define authorizations that not only select which service providers can access location/profile information, but can also constrain the area and time in which they can send their offers to the users. The LTS is in charge of enforcing the authorizations. Among other efforts, the IETF Geopriv working group is proposing a format for expressing privacy preferences for location information [6]. With respect to the threats identified in the previous section, access control is an important component of a privacy preserving solution for all the threats. However, the best results in addressing the threats would probably be obtained by coupling access control with the anonymization techniques discussed below.

B.2 Temporal and Spatial Generalization The threats s.1b and s.2b illustrated in Section A.1 have many analogies with the problem of guaranteeing anonymity of personal data extracted from a relational database (see,

P

e. g., [15]). Typical solutions involve either the de-identification of data, essentially avoiding the presence of values that may directly or indirectly identify the user, the obfuscation of sensitive data, or the separation of identifying values from sensitive data. The first two solutions are usually based on the generalization or suppression of attribute values. Despite there are specific issues that distinguish the location-based problem from the analogous one in the relational database scenario, similar techniques can be applied. Indeed, the dynamic change of spatio-temporal resolution is an obfuscation technique based on generalization.

The idea of adapting spatio-temporal resolution to provide a form of location k-anonymity can be found in [8]. In the field of relational databases, a tuple is said to be k-anonymous if, considering the values of the attributes that could lead to re-identification, it is indistinguishable from other $k-1$ tuples. Analogously, an LBS request is considered k-anonymous if in the same area and temporal interval of the request there are k users that could have submitted that request. The privacy preserving technique consists of enlarging the location area and time interval of a request in order to include $k-1$ other potential users. This work is extended in [7,11,13]. The proposals in [7,13] support the use of a different value of k for different users. In [7], a slightly different notion of k-anonymity is used: the authors require the other $k-1$ users to have actually sent a request. In [13], each user can also specify the parameter $A_{\min}$ that indicates the minimum size of the area that the LTS should forward to the SP. In [11], a possible problem with the generalization proposed in [8] is pointed out and a solution is presented. The notion of k-anonymity in LBS has been more formally defined in [4], where, as in [8], the only requirement is the presence in the same spatio-temporal context of $k-1$ *potential* senders, which is a much weaker requirement.

The application of the techniques proposed in [7,8,11,13] avoids the association by an attacker of the identity of a request sender with a group of identities smaller than k. Hence, the addressed threat is s.2b and indirectly s.1b. The proposed solution partially solves the problem. Indeed, by applying this privacy preserving technique, it is ensured that STdata cannot be used to infer the identity of the request sender. However, the case in which a combination of values of STdata and SSdata is used is not considered.

The proposed solution ensures anonymity of the request sender even in the case in which the attacker knows the location and the identity of every person. If this very conservative assumption is adopted, it is always possible for an attacker to associate the identity of a user to their STdata (threat s.1) even if no requests are submitted. Therefore, in this case, the scope of the technique is limited to the prevention of threat s.2b.

B.3 Identification and Prevention of Critical Request Traces An important aspect in the dynamic scenario is how an attacker can identify a request trace and how a privacy preserving system can avoid it. Two cases have been considered.

In [1,2], LBS's that require a pseudo-id are considered. The proposed privacy preserving technique is based on the notion of *mix-zone* introduced by the authors and aims at avoiding the instance that an attacker traces the requests from the same user for a long period of time. The central idea is to change a user's pseudo-id each time the user enters a mix-zone. A mix-zone is analogous to a mix-node in communication systems [5] and can be intuitively described as a spatial area such that, if an individual crosses it, then it won't be possible to link his future positions (outside the area) with known positions (before entering the area). Here, "link" defines the association of different requests to a single trace.

The results can be applied in the dynamic scenario, but cannot be used to provide a complete solution to any of the threats described in Section A.2. Indeed, the technique aims at reducing the length of the request traces but does not evaluate if sensitive information is released. Nevertheless, reducing the request trace length is an important task that facilitates privacy protection in a dynamic context. Hence, this technique could be very useful as a part of a privacy preserving system (like in [4]).

A different approach to the issue of request traces is to consider LBS's that do not require pseudo-ids. This case is considered in [9], where the authors experimented to see if it is possible for an attacker to trace a user. A known algorithm for tracking multiple objects is applied to trace a small number of users whose locations are frequently collected. The authors concluded that it is practically possible for an attacker to obtain request traces even if pseudo-ids are not submitted to the SP. This paper does not propose a solution to preserving privacy but is a preliminary step in the definition of a technique that could be used by a privacy preserving system to evaluate if a user is possibly being traced by an attacker. In the absence of such a solution, a privacy preserving system should adopt a conservative approach assuming that a user can always be traced.

B.4 Techniques to Prevent Location Identification In [10], it is assumed the case in which each user specifies the *sensitive areas*, i. e., geographic positions that should never be associated to the presence of that specific user. The aim of the proposed privacy preserving system is to avoid that an attacker can understand that the user visited a sensitive area. The straightforward solution of suppressing all the requests from these areas is not effective since an attacker could infer that a user visited a sensitive area only from her request trace outside the sensitive area. This situation is analogous to the one used for threat d.4.

The proposed solution is based on a partitioning of all the areas (sensitive or not) in zones such that each zone includes at least k sensitive areas. Then, each request is suspended until the user crosses a zone boundary. If the user has not visited a sensitive area, all the pending requests are submitted, otherwise they are suppressed.

To our knowledge, the proposed technique is the first one that addresses this kind of problem. However, it is not clear if it is an effective solution. First, it is debatable if it is appropriate to extend the concept of k-anonymity to sensitive areas. Indeed, if a user specifies some sensitive areas, she does not want her identity to be associated with any of them; On the contrary, the proposed solution allows an attacker to infer that a user visited a sensitive area even if it cannot say which one it is in a set of k. Secondly, it is not clear if it is acceptable to always postpone the submission of a request until a user changes a zone. Finally, the way in which the zones are constructed is critical. Indeed, it seems possible in some specific cases that an attacker could infer the exact sensitive area a user visits.

B.5 *k*-Anonymity Techniques in a Dynamic Scenario

The extension of location k-anonymity to the dynamic scenario has been investigated in [4]. The investigation presents a basic question. Is a trace k-anonymous if each request in the trace is k-anonymous according to the definition for the static scenario? The answer is negative; indeed, if m requests constitute the trace, the attacker may have available m sets, each one with at least k candidate individuals. However, since he knows that all the requests in the trace were made by the same individual, he can perform the intersection of the m sets, possibly obtaining less than k individuals.

The formal property needed to guarantee k-anonymity of a trace of LBS requests is called *historical k-anonymity*. Some preliminary definitions are necessary to formally define it. It is reasonable to assume that the LTS not only stores in its database the set of requests that are issued by each user, but also stores for each user the sequence of her location updates. This sequence is called *Personal History of Locations* (PHL). More formally, the PHL of user U is a sequence of 3D points $(\langle x_1, y_1, t_1\rangle, \ldots, \langle x_m, y_m, t_m\rangle)$, where $\langle x_i, y_i\rangle$, for $i = 1, \ldots, m$, represents the position of U (in two-dimensional space) at the time instant t_i.

Note that a location update may be received by the LTS even if the user did not make a request when being at that location. Hence, for each request r_i there must be an element in the PHL of the user issuing r_i, but the vice versa does not hold. This has an intuitive motivation in the fact that the anonymity set for a certain area and a certain time interval is the set of users who were in that area in that time interval and who could *potentially* make a request.

A PHL $(\langle x_1, y_1, t_1\rangle, \ldots, \langle x_m, y_m, t_m\rangle)$ is defined to be *LT-consistent* with a set of requests $r_1, \ldots, r_n$ issued to an SP if, for each request r_i, there exists an element $\langle x_j, y_j, t_j\rangle$in the PHL such that the area of r_i contains the location identified by the point x_j, y_j and the time interval of r_i contains the instant t_j.

Then, given the set R of all requests issued to a certain SP, a subset of requests $R' = \{r_1, \ldots, r_m\}$ issued by the same user U is said to satisfy *Historical k-Anonymity* if there exists $k-1$ PHLs $P_1, \ldots, P_{k-1}$ for $k-1$ users different from U, such that each $P_j, j = 1, \ldots, k-1$ is LT-consistent with R'.

In practice, it is clearly difficult to keep request traces k-anonymous, thus, techniques like mix-zone or frequent change of pseudo-id reducing the length of traces are indeed very important. In [4], a preliminary investigation is reported on the techniques that could be used to preserve historical k-anonymity.

Key Applications

LBS's have been extensively used both in military as well as in commercial applications. With the diffusion of location-aware devices and with the increasing precision of localization techniques, many new commercial applications based on LBS's are being deployed. These applications are not only targeted to business users, but also to generic users interested in LBS for their spare time activities.

LBS privacy preserving techniques can be implemented in a middleware layer to provide privacy protection for most of these applications. Clearly, the level of protection for a given application and for a given user category will be driven by different parameters and possibly with different techniques.

Moreover, the assumptions on the available external knowledge will also determine the most appropriate features that the privacy protection middleware should have. As presented in Section A, the functionality of such a middleware can be focused on protecting the association of the user identity with

- location or trace information,
- service access information,
- both of the above.

Future Directions

Different research directions can be identified in this field.

- **Definition of a formal framework.** Existing approaches would benefit from a rigorous formalization of problems and solutions; In particular, formal definitions of

P

privacy threat and *defense technique* should be provided. The formalization should also be sufficiently expressive, modeling different kinds of knowledge that may be available to the attacker. Indeed, existing approaches do not explicitly define the knowledge that may be available to the attacker; consequently, the proposed defense techniques are subject to critique based on counterexamples, assuming that there is specific knowledge available to the attacker.

- **Design of safe defense techniques.** Based on the formal characterization of a privacy problem either in the static or in the dynamic case, new defense techniques should be designed for that specific problem. Each defense technique should be safe with respect to the specific assumptions made about the knowledge that may be available to the attacker. Moreover, defense techniques should be optimized with respect to the global processing costs at the LTS.
- **Collection and/or generation of significant data for experiments**. The applicative and critical nature of this research field makes it crucial to verify the effectiveness of the proposed defense techniques with real-world data. However, real-word data should include the movement traces of a large number of users for a long period of time and a significant collection of sensitive service requests that these users have issued. Obtaining this information or generating it with a sufficiently accurate model is a challenging task.

Acknowledgements

This work was partially supported by Italian MiUR InterLink project N.II04C0EC1D, and the US NSF grants IIS-0430402 and IIS-0430165.

Cross References

► Indexing, Hilbert R-tree, Spatial Indexing, Multimedia Indexing

Recommended Reading

1. Beresford, A.R., Stajano, F.: Location Privacy in Pervasive Computing. IEEE Pervasive Computing **2**(1):46–55 January–March (2003)
2. Beresford, A.R., Stajano, F.: Mix zones: User privacy in location-aware services. In: PerCom Workshops, pp. 127–131. IEEE Computer Society (2004)
3. Bettini, C., Mascetti, S., Wang, X.S., Jajodia, S.: Anonymity in Location-based Services: Towards a General Framework. In: Proc. of the 8th International Conference on Mobile Data Management (MDM). IEEE Computer Society (2007)
4. Bettini, C., Wang, X.S., Jajodia, S.: Protecting privacy against location-based personal identification. In: Jonker, W., Petkovic, M. (eds.) Secure Data Management, Lecture Notes in Computer Science, vol. 3674, pp. 185–199. Springer (2005)
5. Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM **24**(2):84–88 (1981)
6. Cuellar, J., Morris, J., Mulligan, D., Peterson, J., Polk, J.: Geopriv requirements. RFC 3693, IETF (2004)
7. Gedik, B., Liu, L.: Location privacy in mobile systems: A personalized anonymization model. In: ICDCS, pp. 620–629. IEEE Computer Society (2005)
8. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: Proc. of the First International Conference on Mobile Systems, Applications, and Services (MobiSys), USENIX (2003)
9. Gruteser, M., Hoh, B.: On the anonymity of periodic location samples. In: Hutter, D., Ullmann, M. (eds.) SPC, Lecture Notes in Computer Science, vol. 3450, pp. 179–192. Springer (2005)
10. Gruteser, M., Liu, X.: Protecting privacy in continuous location-tracking applications. IEEE Security & Privacy **2**(2):28–34 (2004)
11. Kalnis, P., Ghinta, G., Mouratidis, K., Papadias, D.: Preserving anonymity in location based services. Technical report TRB6/06, University of Singapore (2006)
12. LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Incognito: efficient full-domain k-anonymity. In: SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, pp. 49–60. ACM Press (2005)
13. Mokbel, M.F., Chow, C., Aref, W.G.: The new casper: query processing for location services without compromising privacy. In: Proc. of the 32nd International Conference on Very Large Data Bases (VLDB), ACM (2006)
14. Myles, G., Friday, A., Davies, N.: Preserving Privacy in Environments with Location-Based Applications. IEEE Pervasive Computing **2**(1):56–64 January–March (2003)
15. Samarati, P.: Protecting respondents' identities in microdata release. IEEE Transactions on Knowledge and Data Engineering **13**(6):1010–1027 (2001)
16. Youssef, M., Atluri, V., Adam, N.R.: Preserving mobile customer privacy: an access control system for moving objects and customer profiles. In: MDM '05: Proceedings of the 6th international conference on Mobile Data Management, pp. 67–76, New York, NY, USA, ACM Press (2005)

Private Visual Thinking; Reexpression

► Exploratory Visualization

Probabilistic Map Algebra

► Bayesian Network Integration with GIS

Probability Networks

► Bayesian Network Integration with GIS

Probability Theory

► Objects with Broad Boundaries
► Uncertain Environmental Variables in GIS

Probability Treshold Indexing

► Spatial Data, Indexing Techniques

Probe Vehicle Data

► Floating Car Data

Problem of Seven Bridges of Konigsberg

► Graph Theory, Konigsberg Problem

Process

► Geographic Dynamics, Visualization And Modeling
► Temporal GIS and Applications

Process Model

► Hierarchical Spatial Models

Processes and Events

ANTONY GALTON
School of Engineering, Computer Science,
and Mathematics, University of Exeter, Exeter, UK

Synonyms

Activities and occurrences

Definition

The terms 'process' and 'event' both refer to things that happen or go on in time, and are invariably associated with some kind of change. The terms are often used loosely, sometimes interchangeably, and there is little consensus on how they should be defined or distinguished from one another. The following definitions are suggested to bring the terms into line with their main everyday uses while providing a useful basis for technical discussions.

Process: An ongoing dynamic situation involving the activity of one or more material things or portions of matter. Key properties of processes are that (1) they are conceptualized as homogeneous, i. e., if a process is going on over some interval of time then it is also going on over all the subintervals of that interval (at least down to some minimal length associated with the inherent 'grain-size' of the process), and (2) they are open-ended, i. e., if a process is going on at a particular time then in principle it can continue going on into the future—thus processes are not intrinsically bounded by an end-point or completion.

Event: An individual occurrence or episode singled out from the ongoing processual flux, with a definite beginning and end (which coincide in the case of instantaneous events). Key properties of events are exactly opposed to those of processes: (1) they are not homogeneous, since if an event occurs over an interval then it does not occur over any of the subintervals (so the event's occurrence is a global property of the interval, not a local property of its subintervals), and (2) events are not open-ended—once an event has occurred it cannot go on occurring (although another event of the same type may begin).

Historical Background

Questions about the general classification of processes, events, and related categories have been pursued in three main areas: philosophy, linguistics, and artificial intelligence. The philosophical contribution is largely in the areas of metaphysics, ontology, and the theory of action. In linguistics, the focus is on classifying verbs and verb-phrases in order to explain the different ways in which they are used, and in particular to account for their interaction with temporal elements such as tense and aspect, temporal prepositions, and so on. In artificial intelligence, the emphasis is on representing knowledge about processes and events in such a way as to facilitate reasoning about them for purposes such as planning, prediction, and explanation.

On the philosophical background, a useful point of entry is provided by [2], which collects together a wide range of papers on events. In linguistics, [6,7] provide useful background on the classification of expressions denoting processes and events. In artificial intelligence, two classic papers are [1,5]. A useful collection of papers which includes work from all three background areas is [4].

In GI Science, while the necessity for handling temporal phenomena has been acknowledged for some time now, progress has been hampered by the lack of principled ways of describing these phenomena; in particular, the failure to distinguish adequately between processes and events has led to a certain amount of confusion in the literature.

P

As Worboys [10] says, 'One person's process is another's event, and vice versa'. It is important not to let the terminological variation detract from the real merit of much of the work that has dealt with dynamic phenomena in GIS, but it remains true that progress would be expedited if a common usage of the terms could be agreed.

Scientific Fundamentals

Any application which handles both processes and events should also handle the relationships between them. Some of these relationships are very general, belonging at a high level of abstraction; such relationships can be systematized into an *algebra* of processes and events. The operations of such an algebra include ways of deriving events from processes and ways of deriving processes from events. An example of the former would be an episode of erosion leading to the ultimate separation of a promontory from the mainland; an example of the latter would be flow of traffic past a junction, which is a process consisting of many individual events corresponding to the passage of individual cars past the junction.

The open-ended, homogeneous nature of processes means that processes should be regarded as existing from moment to moment, in contrast to an event which spans a whole interval and is not present as a whole at individual moments within the interval. It makes sense to say that a process is in operation at a single time of observation (even though certain *methods* of observation, such as the capturing of a 'snapshot' image, may fail to reveal the processes in operation), whereas events are normally identified retrospectively by synthesizing a sequence of states over an interval or comparing some initial state of affairs with an eventual outcome. Thus a process like the flow of a river can be observed directly, and its properties (e. g., its speed) measured at the moment of observation. A corollary of this is that processes can change: the properties of a process measured at a later time may differ from those measured earlier. This in turn allows the definition of second-order processes such as the increase in flow of the river.

For events, by contrast, a different, historical perspective is adopted. An event is a discrete unity[1] that is located at a particular time, either a point or an interval, so it does not make sense to ascribe change to it: rather, it just happens and then so to speak 'sits there' in the historical record. (In cases where it seems as though an event is changing, it is invariably the constituent process of the event, rather than the event itself, to which change is more properly ascribed—e. g., to say that a volcanic eruption became more violent is to say that the erupting process changed in this way; the eruption as an event is a unitary whole which subsumes the profile of variable violence within its temporal 'shape' but cannot meaningfully be said to undergo change itself.)

Note however the word process is often used to refer to what are in fact *procedures*, e. g., the process of refueling an aircraft. This is something that follows a prescribed pattern, involving various events and processes; it is typically something for which explicit instructions could be given. Each individual instance of a procedure being carried out is an event, not a process in the sense used in this article. Although the word 'procedure' suggests something involving human agency, similar 'structured events', proceeding according to a fixed pattern, occur in nature, although perhaps more typically in biological than geographical contexts, e. g., cell division.

With both processes and events, it is necessary to distinguish generic *types* from specific individual *instances* of those types. This is particularly clear in the case of events: there is an obvious difference between the notion of a volcanic eruption as a generic type of event that may occur at many different places and times, and specific instances of this type such as the eruption of Vesuvius in 79 A.D. or that of Mount St Helens on May 18th 1980. This distinction is sometimes obscured by normal ways of speaking: 'the same thing happened again' does not refer to the same individual event but to another individual of the same *type*. With processes, similarly, longshore drift as a generic process-type can be distinguished from the current realization of that process along a particular stretch of coastline.

Processes can be broadly classified as steady-state, cyclic, irregular, or progressive, depending on the profile of their temporal development.

A steady-state process maintains an existing state of affairs: it can operate over arbitrarily long periods while producing no net change. An example is the flow of water through a lake, where the inflow is exactly balanced by the outflow, resulting in the water-level remaining constant; although there is undeniably a process in operation here, to an outside observer it looks static. Even where movement is discernible, a process can still be classed as steady-state if there is no overall change, at a certain level of description: the flow of traffic along a stretch of motorway, for example, can be regarded as steady-state so long as the speed and density of the traffic remain more or less constant.

In a cyclic process, there is a regular periodic variation in the associated state of affairs. An example is the rise and fall of the tides along a shoreline, endlessly repeating the cycle of high tide followed by low tide followed by high tide again. The periodicity does not have to take a simple sinusoidal form: several cycles can be superposed to produce more complex profiles of variation. This is true of the

[1] The quality or state of not being multiple, i. e. oneness.

tidal process, since the half-daily alternation of high and low is accompanied by a monthly alternation of spring and neap as well as longer-term alternations resulting from the complex interaction of the relative motions of the earth, moon, and sun.

Sufficiently complex superpositions of periodicities can result in what appears to be a completely irregular process profile, and of course it becomes a matter for scientific investigation as to whether an apparently irregular process is, in fact, the product of some complex combination of regular cycles. Many meteorological phenomena are of this kind, which is what makes them hard to predict.

The final category of processes are those which may be described as progressive, in which the operation of the process results in a cumulative change in some state of affairs. The process of urbanization by which some built-up area steadily encroaches on what was open country is progressive in this sense: the process does result in net change, and the longer the process operates for, the greater the magnitude of the change.

How a process is classified can depend critically on the temporal 'grain size', or granularity, under which it is described. Many apparently steady-state processes appear on closer examination to be cyclic or irregular. If the minimal temporal interval under consideration is greater than half a day, the high tide/low tide periodicity becomes indiscernible, but the longer term spring tide/neap tide periodicity may remain. Again, a steady-state traffic flow, when described at a finer temporal granularity, consists of the passage of an endless succession of individual vehicles, and this may be viewed from a certain point of view as a cyclic or irregular process.

Similarly a process which appears merely cyclic under a fine temporal granularity can be revealed as progressive when the focus is shifted to longer time periods. The flow of traffic along a particular stretch of road is cyclic to the extent that it undergoes regular and predictable daily and seasonal variation; superimposed on this there will be many smaller-scale irregularities, but more significantly, over a longer period of many years there may be a progressive increase in traffic levels. This might be understood as one process which has steady-state, cyclic, irregular and progressive aspects depending on the temporal 'focus' under which it is described, or as the superposition of a number of separate processes operating over different time scales. The former view is more concrete in the sense that the process simply is the flow of traffic that is observed; under the latter view the processes become more abstract and only accessible by inference from the observed flow.

None of the above applies to events. It is true that an event can be described as periodic, but in this case what is meant is that there is a process consisting of the regular repetition of some event type: each individual repetition is one event, and as such is not itself periodic.

Events may be classified as *punctual* (i. e., point-like) or *durative* (i. e., extended or interval-like) depending on whether or not they have appreciable duration. This too is a matter of granularity: on a scale of hours or days, a volcanic eruption may be an extended affair, but from a perspective spanning many years it may be effectively instantaneous.

Events may also be classified in terms of whether they are characterized *internally* in terms of their constitution, either as a 'block' of some process or as a composite of two or more subevents, or *externally* in terms of their effects, e. g., an event may be defined in terms of a transition between specified start and end states. A volcanic eruption is internally characterized as an episode which consists of the eruption process operating at a particular location over some particular interval of time. The separation of Great Britain from the European mainland is externally characterized as the transition from a state of affairs in which Britain and Europe formed part of a single land mass to one in which they were separated by the sea. An event which is externally characterized will usually in fact come about as a result of the operation of some process; but the external characterization does not explicitly refer to this. It is also possible to describe an event using both internal and external characteristics: the event reported in the statement 'John walked to the station' is characterized externally by the fact that it is a transition from a state of affairs in which John is not at the station to a state of affairs in which he is; and it is characterized internally by the fact that it comes about as a result of John's spending a while walking.

Key Applications

Processes, by nature, involve some form of change; in many cases, the change in question is motion, and it is usually continuous (or, if discrete, is only revealed to be so at a finer temporal granularity than that at which it is being represented as a process). The application areas in which the notion of process is likely to be invoked are therefore those in which there is change that is effectively continuous. This has implications for the kinds of technologies that can be used for modeling or otherwise representing processes.

The simplest way of handling time and change in the context of GIS is by means of a sequence of static "snapshots" representing the world at times $t_1, t_2, t_3, \ldots$. Assuming that individual snapshots contain no information about processes that are in operation at the times they represent, the existence of processes can only be inferred by tracking the changes between successive snapshots. In effect this presents a sequence of events of the form 'between t_i and t_j the world changed from this state to that state'. These events by nature form a discrete series, which does not do justice to the continuous nature of processes. The processes present in the world (which give rise to the events, as it were forming their temporal substance), are not visible in such a model.

To represent processes explicitly, therefore, some additional technology is needed. One possibility is to include processes as first-class objects within individual snapshots: in effect, to treat a process as an individual entity which may be in operation from one snapshot to the next, itself perhaps undergoing change. An alternative approach is to move beyond the snapshot model and present the course of history directly, with all entities—processes, events, and ordinary objects—modeled as four-dimensional 'chunks' of reality. One can then obtain different views of the world by segmenting these chunks in different ways; for example, a conventional snapshot is now a 'slice' of the four-dimensional world orthogonal to the time axis, whereas the history of an object encapsulates the changes it undergoes in a 'life-line' that is extended along the time dimension. As yet there is no clear consensus as to how best to handle events and processes in the context of GIS, and it is therefore not possible to point to an established way of proceeding.

Since processes and events are ubiquitous in the world, almost any GIScience application may need to take them into account. The following paragraphs present a sample of areas where this need has been felt particularly keenly, and which have therefore figured prominently in recent discussions about how the temporal dimension should be integrated into GIS.

Weather systems. Yuan [11] distinguishes a precipitation event (i. e., 'the occurrence of precipitation in the study area') from its constituent precipitation processes (which describe 'how it rains'). This is broadly in line with the way in which the relationship between processes and events was characterized above. Yuan describes algorithms for assembling precipitation processes and events from raw precipitation data and for computing information about their behavior and interaction. This approach is advocated to provide enhanced support for 'complex spatio-temporal queries on the behavior and relationships of events and processes'.

Coastal Geomorphology. The sea coast is an excellent place to see both processes and events in action. The breaking of a particular individual wave is an event, but the regular succession of waves breaking along the shoreline is a process. On closer analysis, individual wave-breaks can be seen to be made up out of various kinds of processes, and if different types of wave are distinguished according to how they break (e. g., 'spilling' and 'plunging') then this explicitly invokes different constituent process types. In the longer term, erosion and deposition phenomena constitute processes which, in aggregate, may result in discrete events such as the separation of a headland from the mainland or the formation of a spit. Methods of representing such phenomena in an information system are described by [9].

Oceanography. Often the focus of interest is the profile of a process: its increases and decreases, its peaks and troughs, or periods of stability. The general picture assumed here is as follows: the raw data, as delivered by various sensors, takes the form of sequences of quantitative values of some time-varying observable. An example would be quantities such as the speed and direction of winds and current, wave height, air temperature and so on that might be measured by a buoy tethered to the ocean floor at a particular location. Ongoing changes in any of these quantities are processes. Salient demarcated episodes in the evolution of any of these process are events.

In the GoMOOS (Gulf of Maine Ocean Observing System) project (see http://www.gomoos.org/), a number of buoys at different locations across the Gulf of Maine are making regular observations of key oceanographic variables; the problem is then how to derive from these observations an understanding of the large-scale oceanic processes operating in the gulf. To facilitate such understanding it is essential to have an appropriate ontology for classifying processes and events, refined by careful consideration of the spatial dimension.

Traffic systems. Worboys [10] uses algebraic methods to specify traffic flow at a four-way stop. This is, essentially, an open-ended process built up from a sequence of atomic events. The atomic events are of two kinds: a vehicle arriving at the intersection, and a vehicle moving forward across the intersection. The specification shows how the ongoing process can be generated from an appropriate sequencing of such events. This describes the normal pattern of events, but in addition the modeling of traffic systems should take into account events such as traffic jams and accidents which arise from the particular conditions of the traffic flow process.

Traffic flow can be regarded as a process which exists at all points on the network at every time. It may be characterized by various attributes, of which the most important are

- speed (measured in units of distance/time)
- density (measured in units of vehicles/distance)
- rate of flow (measured in units of vehicles/time).

These attributes vary both spatially across the network and temporally. Each of the attributes has a value at each position and time, the rate of flow being the product of speed and density.

One thing which makes traffic flow particularly interesting is that it may be viewed from two quite different kinds of perspective, both of which will be important in different contexts. On the one hand, there is the bystander's point of view, that is, the point of view of someone positioned at a particular point in the network and watching the traffic flow past. This is the point of view that is relevant to a pedestrian trying to cross the road, or to a speed camera in a fixed position beside the road. It is also the point of view of someone living next to the road and concerned about traffic noise, the safety of their children, etc. The process that is of concern here is the traffic flow *past a point*; What the bystander observes is the fluctuation in the attributes of this process.

A different point of view is that of the participant in the traffic flow itself, i. e., the drivers of the vehicles whose motion through the network constitutes the flow. The flow as viewed from a given vehicle can be regarded as a process in its own right, whose time-varying attributes may be quite different from those of the process viewed by the bystander. To illustrate this, consider a long stretch of road with a single intersection half-way along, controlled by traffic lights. Then someone stationed at a point close to the traffic lights will observe a cyclical variation in the flow along this road, but the driver of a vehicle traveling along the road may, depending on the state of the lights when he reaches the intersection, either observe a uniform flow, or a flow which comes to a halt on one occasion and then a little while later starts up again. It is important that what is seen by each of these observers is a perfectly good process; and clearly there are systematic relationships between the processes seen by stationary observers at different points in the network and the processes seen by moving observers participating in the flow.

Future Directions

Temporal GIS is still an emerging field, and as such has not yet arrived at an established consensus on the terminology and classification of temporal phenomena—including broad categories such as process and event. As a result, development of temporal capability in GIS has mostly proceeded in a piecemeal and *ad hoc* fashion, with no solid basis in underlying theory. On the other hand, such theorizing as exists has tended to be divorced from the practicalities of implementing useful temporal functionality in GIS. A key desideratum for the future is therefore to establish an appropriate theoretical basis for describing the temporal dimension that is firmly grounded in practicality.

One element in particular will need to be looked at: in geographical information science it is customary to distinguish between object-based approaches, which think of space as populated by a collection of discrete objects each with a unique identity and attributes, and field-based approaches, which think rather of variables taking different values at different locations, providing a coverage of space. The ontologies implicit in most formal representation languages are heavily biased towards the object-based view of the world, but in application areas such as oceanography, meteorology, and coastal geomorphology a field-based view may be more appropriate. For this reason, any adequate treatment of processes and events must cover not only the changes undergone by individual objects, but also the the phenomena that may be observed in continuously varying fields. Indeed, as the examples above show, it is precisely in this latter kind of example that a proper treatment of processes is most urgently needed.

Cross References

► Geographic Dynamics, Visualization And Modeling
► Movement Patterns in Spatio-temporal Data
► Patterns in Spatio-temporal Data
► Sequential Patterns, Spatio-temporal
► Temporal GIS and Applications

P

Recommended Reading

1. Allen, J.: Towards a general theory of action and time. Artif. Intell. **23**, 123–54 (1984)
2. Casati, R., Varzi, A. (eds.): Events. The International Research Library of Philosophy. Dartmouth, Aldershot, UK (1996)
3. Hornsby, K., Worboys, M. (eds.): Event-oriented Approaches in Geographic Information Science. Lawrence Erlbaum Associates. Special issue of: Spatial Cognition and Computation, vol. 4, number 1 (2004)
4. Mani, I., Pustejovsky, J., Gaizauskas, R. (eds.): The Language of Time: A Reader. Oxford University Press, Oxford (2005)
5. McDermott, D.: A temporal logic for reasoning about processes and plans. Cogn. Sci. **6**, 101–55 (1982)
6. Moens, M., Steedman, M.: Temporal ontology and temporal reference. Comput. Linguist. **14**, 15–28 (1988)
7. A. P. D. Mourelatos. Events, processes, and states. In: Tedeschi, P., Zaenen, A. (eds.) Tense and Aspect, pp. 191–212. Academic Press, New York (1981)
8. Peuquet, D.J., Duan, N.: An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. Int. J. Geogr. Inf. Syst. **9**, 7–24 (1995)
9. Raper, J.: Multidimensional Geographic Information Science. Taylor and Francis, London and New York (2000)
10. Worboys, M.F.: Event-oriented approaches to geographic phenomena. Int. J. Geogr. Inf. Sci. **19**(1), 1–28 (2005)
11. Yuan, M.: Representing complex geographic phenomena in GIS. Cartogr. Geogr. Inf. Sci. **28**, 83–96 (2001)

Profile, Topological

► Spatial Data Transfer Standard (SDTS)

Profiles

► Spatial Data Transfer Standard (SDTS)

Programmable GIS Components

► MapWindow GIS

Progressive Approximate Aggregation

IOSIF LAZARIDIS, SHARAD MEHROTRA
Department of Computer Science,
University of California, Irvine, CA, USA

Definition

Aggregate queries are answered by computing a single scalar value over a set of relevant data objects, e. g., the average temperature over all sensors in a region of space. Often, the precise value is not needed because the user may not be interested in knowing the temperature to the last decimal point. Additionally, the mode of visualization of the answer, e. g., in a virtual navigation, may in itself impose restrictions by necessitating either a high frame rate or the color-coding of the answer at a specific resolution. Progressive approximate aggregate queries compute the answer progressively, coming up with an initial estimate and refining it until the time deadline (e. g., time to render the frame), or answer quality precision (e. g., $\pm 1°C$) is reached. Thus, they are a flexible way of query processing since they make no assumptions about the time/accuracy specifications imposed by the user application and can accommodate a wide variety of such specifications in a unified way.

Main Text

On-line aggregation was proposed as an approximate aggregation mechanism. Data objects are read and the aggregate function is computed with them. Using statistical techniques, probabilistic confidence intervals on the answer are computed; the user can stop the query whenever he is satisfied with the answer quality.

Progressive approximate aggregate queries using Multi-Resolution Aggregate tree (MRA-tree) data structures are another way of progressively approximating the answer. They avoid calculating the aggregate over individual tuples since these aggregates are pre-computed and stored in nodes of the MRA-tree. Moreover, at each step they provide a point estimate of the aggregate value (e. g., 34C) and a deterministic interval of confidence (e. g., the interval [33, 36]) guaranteed to contain the exact answer. Progressive approximation techniques are a powerful aid for geospatial visualization systems incorporating large amounts of data or requiring fast query answering, allowing such systems to function efficiently and to produce high quality answers with provable bounds using the available system resources.

Cross References

► Aggregate Queries, Progressive Approximate
► Multi-Resolution Aggregate Tree

PROJ

► Open-Source GIS Libraries

Properties, Geometric

► Geography Markup Language (GML)

Properties, Simple

► Geography Markup Language (GML)

Property Register

► Cadastre

Protective Action Zone

► Emergency Evacuations, Transportation Networks

Provable Properties

► Geosensor Networks, Formal Foundations

Proximity Matrix

► Spatial Weights Matrix

PR-Quadtree

► Quadtree and Octree

PSS

► Geocollaboration

Public-Domain Software

► deegree Free Software
► PostGIS

Public Health

► Pandemics, Detection and Management

Public Health and Spatial Modeling

ANDREW B. LAWSON
Arnold School of Public Health,
University of South Carolina, Columbia, SC, USA

Synonyms

Mapping; Spatial models; Disease mapping

Definition

In this chapter the emphasis is on the analysis of health data which are geo-referenced. Often data of this kind arise in the analysis of clustering of disease, or in the estimation of excess or relative risk in small administrative regions. Another aspect of this analysis may be the need to assess the relation between the spatial location of cases of disease and covariates, often spatially distributed. This is often called ecological analysis, especially when a change of spatial scale is involved. Finally, the possibility of carrying out prospective surveillance of disease maps is considered.

Historical Background

Spatial modeling of Public Health data is a wide subject and encompasses a range of topics where the geographical or spatial distribution of disease is of importance. In what follows there will be an emphasis on disease analysis as opposed to other aspects of Public Health sciences (such as health services research, health promotion or education). Statistical methods employed in this area are also diverse in their range and besides basic exploratory and descriptive methodology common to many subject areas, there is a need to employ particular *spatial* statistical methods which are designed for such data. The basic characteristic of data encountered in this application area is its *discrete* nature, whether in the form of spatial locations of cases of disease, or counts of disease within defined geographical regions. Hence methods developed for continuous spatial processes, such as Kriging, are not directly applicable or only approximately valid.

Often geographical hypotheses of interest in Public Health focus on whether the residential address of cases of disease yields insight into etiology of the disease or, in a public health application, whether adverse environmental health hazards exist locally within a region (as exemplified by local increases in disease risk). The classic example of the earliest epidemiological study was that of Snow [33] who examined the cholera epidemic in London in relation to water supply pumps. Since the 1970s there has been a growth in interest in geographical analysis of disease and this growth intensified in the 1980–1990 period with the development of geographical information systems (GIS) and public awareness of local environmental risks. The importance of geographical analysis of disease can be easily demonstrated. For example, in a recent study of the relationship between malaria endemicity and diabetes in Sardinia a strong negative relationship was found [4,25], ch 9. This relation had a spatial expression and the geographical distribution of malaria was important in generating explanatory models for the relation. In public health practice it is of considerable importance to be able to assess whether localized areas which have larger than expected numbers of cases of disease are related to any underlying environmental cause. Here spatial evidence of a link between cases and a source is fundamental in the analysis. Evidence such as a decline in risk with distance from the *putative* source of hazard or elevation of risk in a preferred direction is important in this regard.

Scientific Fundamentals

There are four main areas where spatial statistical methods have been developed for Public Health data analysis: relative risk estimation (sometimes simply called Disease Mapping), Disease Clustering, Ecological Analysis, and Disease Map Surveillance. Before looking in detail at each of these areas, it is appropriate to consider some common themes or issues which arise in all areas of the subject. Before considering the study of the spatial distribution of disease, there are some fundamental epidemiological ideas that should be considered.

Relative Risk

Within any geographical area the local density of cases of disease can be studied. There is a desire to examine this as it gives information about local variations in disease. With census tracts then the count of cases of a particular disease could be the data of interest. These crude counts of disease cannot be used on their own as the density of cases will be affected by the variation in the population of the area. This is true whether observing case addresses (the residential address of a disease case) or the aggregated count of cases within small areas.

Hence underlying the disease incidence is the variation in the population 'at risk' of the disease. This background population will vary in its composition (age, gender, susceptibility groups) and in its density with spatial location. Hence this variation should be accounted for in any analysis of the disease occurrence. Clearly, if areas of high susceptibility (with frail population groups) coincide with areas of high disease occurrence then there is likely to be less interest in these areas (in terms of adverse disease presence), than areas where there is high disease occurrence and low number of susceptibles. Local occurrence of disease (counting of cases within areas) within short time spans (e. g. individual months or years) is termed *incidence*. Longer term accumulation of disease cases is often termed *prevalence*. Here the term incidence is used throughout. In general prevalence can be analyzed as for incidence.

To simplify discussion, initially, assume there is a small administrative area (such as a census tract, postcode, zip code, county etc.) within which it is possible to observe the disease incidence. Often there is a desire to compare the observed count of disease with what would have arisen from the underlying population. This will tell us if there is any excess disease risk in the local area. Let's assume there are $i = 1, \dots, p$ tracts or small areas in a study area. Often a ratio of the observed count y_i, in the ith tract to the expected count e_i derived from the background population is used to examine excess risk: the *relative risk* of a disease within the i th area can be estimated by $\frac{y_i}{e_i}$. This ratio represents the relative risk compared to that the local population suggests should be seen in the area. Usually the count y_i will be available from government PH data sources and the expected count (or rate) is usually computed from known rates for the disease in population subgroups (broken by age and gender). This is known as *standardization*. The calculation of expected rates can be very important and different methods of calculation could lead to different conclusions about disease risk. Note that this relative risk definition implies a multiplicative model for risk. This is a common assumption in epidemiology.

Standardized Mortality/Morbidity/Incidence Ratio(SMR or SIR)

The above relative risk ratio is commonly computed for certain types of data. The most common is where incident cases are involved and is called the *standardized incidence ratio* (SIR). Sometimes live cases are described by the term morbidity and so SMR is sometimes used. This can be confusing as when deaths from a disease are recorded (mortality) the same acronym is applied (SMR: standardized mortality ratio). Of course, different expected rate calculations (denominators) would usually be used depending on whether incidence or mortality were to be considered.

Standardization Expected rates in the small areas or tracts $\{e_i\}$ are calculated (estimated) from the local population structure. Usually an external standard population rate will be known and applied to the local population. For example, suppose that the national US rate for prostate cancer (PrCa) is to be used to standardize the rates in South Carolina counties. The rate for different population groups must be known. Hence the rate for each age × gender group must be known nationally and also the population in this groups must be known locally. Define the US rate for PrCa in the kth age group and jth gender group as e_{kj}. Define the population in these groups in the ith area as p_{kji}. Hence the expected rate in the i th area will be simply:

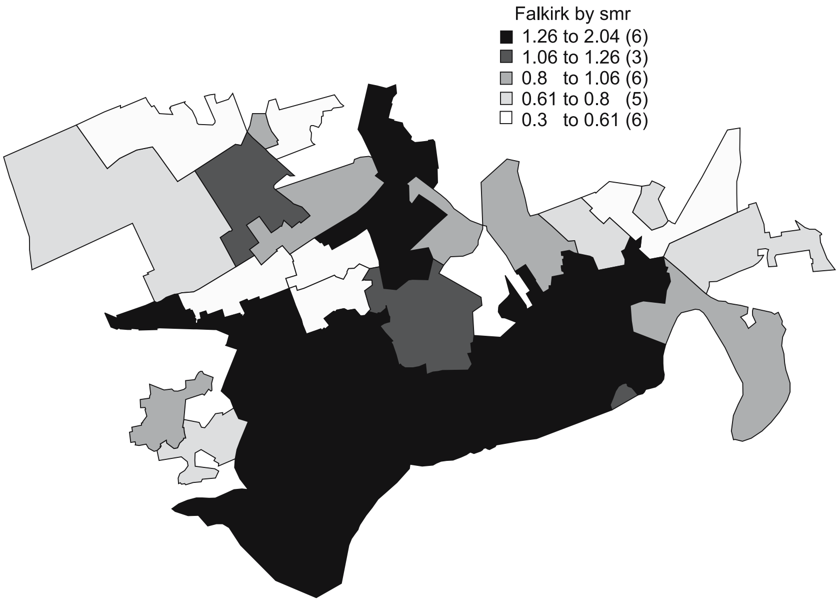
$$e_i = \sum_j \sum_k e_{jk} p_{kji} \,. \tag{1}$$

That is: the numbers in each tract in different age × sex groups are multiplied with known rates for the disease for equivalent groups in a *standard* population. The standard population may be the *national* population (as above) or even the *study region* population. The study region population may be the most relevant if you want to study relative spatial differences across a study region. Note also that other standardizations could be used where covariates are used to standardize the rates.

The standardized ratio of either incidence, mortality or morbidity is the relative risk ratio computed with standardized expected rates, as specified above:

$$SMR_i = \frac{y_i}{e_i} \,. \tag{2}$$

Figure 1 displays a standardized mortality ratio map for 26 census tracts in Falkirk central Scotland. The SMR map is often used by PH professionals to examine the distribution of disease risk. Areas of the map with SMRs greater than 2 or 3 (say) may be of concern. More formally, tests can be carried out to assess whether risk excesses are significant statistically. Visual assessment is not adequate for this purpose. Note also that the SMR is one estimate of relative risk, and there are many other ways to estimate risk.



Public Health and Spatial Modeling, Figure 1 Central Scotland: 26 census enumeration districts (EDs) in the center of the city of Falkirk. Respiratory cancer deaths (SMRs) for the period 1976–1983. Scottish national rate used for standardization

Control Diseases and Expected Rates

Expected rates are commonly used to allow for population effects when count data is observed. Count data is often available readily from government sources. However, for some purposes there is a need to examine the spatial distribution of cases at finer spatial resolutions. Commonly the residential address of cases is the finest level of resolution that can be found. Usually this is only relevant if a small geographic study region is examined. In this case the data form a spatial point process. As for count data, there is a need take population variation into account when examining risk at this spatial resolution level.

Expected rates are usually only available at aggregated geographic scales (census tracts or such like areas) and cant be used effectively at fine resolutions to control for population variations. An alternative is to use the incidence of a *control disease* within the study region. A control disease is matched closely to the risk structure of the disease of interest, but must not display the incidence effect under investigation. For example, live births could be used as a control for childhood leukemia in clustering studies. In that case the address locations of all births would be used as a population surrogate. This leads to two point processes: the leukemia case distribution and the live birth distribution. Of course, live birth is not a disease but in this case is a population indicator. Another example would be the use of residential addresses of coronary heart disease (CHD) as a control disease for respiratory cancer in studies of air pollution. CHD would be thought to affect the same age structure as respiratory cancer, but may be unaffected by air pollution hazards. Of course this may not be a good control as it could be affected by lifestyle variables such as smoking. Lower body cancers (testes, penis, ovaries etc.) has been proposed as a control disease for respiratory tract diseases. These are less affected by inhilatory insults. However care must be taken as some air pollutants can target lower body organs (e. g. nickel). In addition the time lag of disease expression (which is long in the case of cancers) should also be considered.

Note that this control disease is a *geographical* control and is not matched to specific cases. The common feature of each control disease is that it shouldn't be related to the effect of interest. There is some debate about use of these controls as opposed to expected rates from external sources.

The Ecological and Atomistic Fallacy

Many mapping studies attempt to relate incidence of disease in regions with some other measurable *explanatory* variable relating to the etiology of the disease e. g. it may be necessary to examine the relation between the number of smokers in regions and the incidence of respiratory cancer in the same regions. This might be achieved by applying regression analysis to the disease incidence and explanatory variable. The relation between these variables will be statistical and may suffer from the fact that regional totals or averages are used in the assessment of the relation. Hence an average relationship can only be measured. There is no direct link between whether an individual smokes and whether they develop lung cancer.

The *ecological fallacy* arises when such regional average characteristics are ascribed to *individuals* within the region concerned. Any region-based analysis will suffer from this problem. It is known in fact that in some extreme cases the relation between the covariate and the outcome is reversed when individual analysis is carried out. Hence, ecological analyses are some times viewed with caution. Of course, at the aggregate level the relation remains valid. The *atomistic fallacy* occurs when analysis is based on individuals, and the variability of individuals' response to disease is not accounted for in inference at the regional level. These, and other aggregation issues, are further discussed in [15,31,36].

Confounders and Deprivation Indices

All disease maps contain the influence of variables affecting, or pertaining to, the local population which are not accounted for in standardized rates or control diseases. There are two ways to try to allow for these effects:

1. *include as many known explanatory variables in the expected rate or regression model to allow for these effects.* (These variables are called known confounders.)
2. *include the effect of unmeasured confounders via the use of random effects.*

In the first case the solution is to include in the study as many known variables that affect the outcome so that extra variation is explained. Of course it may not be feasible to include all know confounders simply due to (realistic) study limitations. To make allowance for unmeasured confounders (whether known or unknown) it is possible to admit random effects into any regression models. These are additional unobserved variates that will soak up extra variation of various kinds.

Often adverse disease incidence is known to be related to a range of poverty-related explanatory variables e. g. unemployment, housing type,welfare status, car ownership. That is, measurable adverse risk is expected in areas where these variates indicate low income and poverty. These variables are often available from national census. There has been some effort to combine such variables in composite measures known as *deprivation indices* [6]. In North America these are often termed urbanicity indices. Deprivation indices are now routinely available from government census data organizations and can be incorporated directly into a disease map as a covariate or as an offset term.

Some Spatial Statistical Issues

A fundamental feature of geo-referenced data available for analysis in Public Health applications is that it is usually discrete (either in the form of a point process or counting process), and the cases of concern arise from within a local human population which varies in spatial density and in susceptibility to the disease of interest. Hence any model or test procedure must make allowance for this background (nuisance) population effect. The background population effect can be allowed for in a variety of ways. For count data it is commonplace to obtain *expected* rates for the disease of interest based on the age-sex structure of the local population (see e. g. [9], chap. 3), and some crude estimates of local relative risk are often computed from the ratio of observed to expected counts (e. g. standardized mortality/incidence ratios: *SMR*s). For case event data, expected rates are not available at the resolution of the case locations and the use of the spatial distribution of a control disease has been advocated. In that case the spatial variation in the case disease is compared to the spatial variation in the control disease. A major issue in this approach is the correct choice of control disease. It is important to choose a control which is matched to the age-sex structure of the case disease but is unaffected by the feature of interest. For example, in the analysis of cases around a putative health hazard, a control disease should not be affected by the health hazard. Counts of control disease cases could also be used instead of expected rates when analyzing count data.

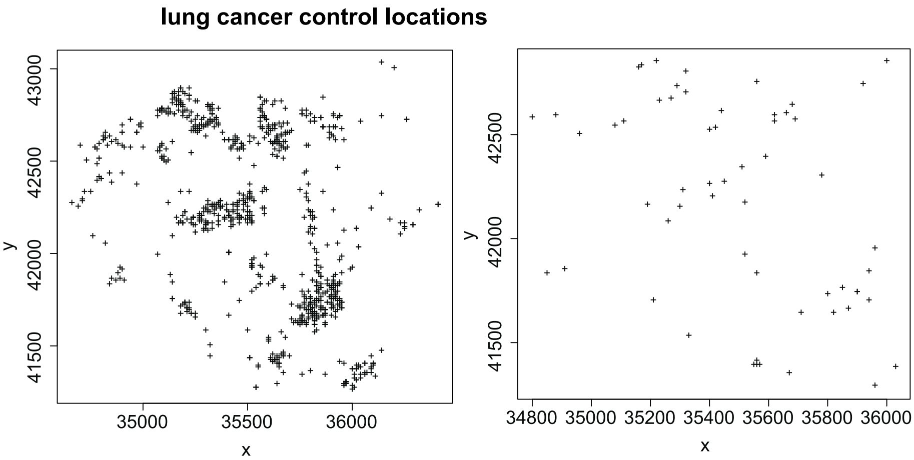
Case Event Data

Figure 2 displays control event (a) and case data (b) maps for a region of the UK for a fixed time period. In this example, larynx cancer case incidence is the distribution of interest while lung cancer distribution is the control disease for the same period.

Case event locations often represent residential addresses of cases and the cases arise from a heterogeneous population that varies both in spatial density and in susceptibility to disease. A heterogeneous Poisson process model is often assumed as a starting point for further analysis. Define the *first-order intensity* function of the case event process as $\lambda(\mathbf{s})$, representing the mean number of events per unit area in the neighborhood of location $\mathbf{s}$. This intensity may be parametrized as:

$$\lambda(\mathbf{s}) = \rho.\lambda_0(\mathbf{s}).\lambda_1(\mathbf{s};\theta) \quad (3)$$

where ρ is the overall rate of the process, $\lambda_0(\mathbf{s})$ is the 'background' intensity of the population at risk at $\mathbf{s}$, and $\lambda_1(\mathbf{s};\theta)$ is a parametrized function of risk. The focus of interest for making inference regarding parameters describing excess risk lies in $\lambda_1(\mathbf{s};\theta)$,, treating $\lambda_0(\mathbf{s})$ as a nuisance function. The function $\lambda_1(\mathbf{s};\theta)$ represents the relative risk measured locally around location $\mathbf{s}$, and $\log\lambda_1(\mathbf{s};\theta)$ is often modeled.



Public Health and Spatial Modeling, Figure 2 Lancashire UK: **a** lung cancer control disease address locations, **b** larynx cancer address locations. Both maps are for the period 1974–1983 and are for incident cases

It is possible that population or environmental heterogeneity may be unobserved in the data set. This could be because either the population background hazard is not directly available or the disease displays a tendency to cluster (perhaps due to unmeasured covariates). The heterogeneity could be spatially correlated, or it could lack correlation in which case it could be regarded as a type of *overdispersion*. One can include such unobserved heterogeneity within the framework of conventional models as a random effect.

This approach can lead to maximum a posteriori estimators similar to those found for universal kriging in geostatistics [23].This approach can also be implemented in a fully Bayesian setting (see e. g. [25] amongst others).

Count Data

Figure 3 displays a typical count data example: congenital death counts for South Carolina counties for the year 1990. A considerable literature has developed concerning the analysis of count data in spatial epidemiology (e. g. see reviews in [3,9,25,26]).

The usual model adopted for the analysis of region counts is to assume that $\{y_i, i = 1, \dots, p\}$ are independent Poisson random variables with parameters $\{\lambda_i, i = 1, \dots, p\}$. Here

$$\lambda_i = \int_{W_i} \lambda(\mathbf{s})\, \mathrm{d}\mathbf{u}\,, \quad i = 1, \dots, p\,,$$

where $\lambda(\mathbf{s})$ is the first order intensity of the underlying cases and W_{i} is the i-th subregion. Often the $\lambda_i s$ are assumed to be constant within areas. Usually the expected count is modeled as

$$E(y_i) = \lambda_i = e_i\theta_i\,, \quad i = 1, \dots, p\,.$$

This model may be extended to include unobserved heterogeneity between regions by introducing a prior distribution for the log relative risks ($\log\theta_i, i = 1, \dots, p$). Incorporation of such heterogeneity has become a common approach and the Besag, York and Mollié (BYM) convolution model is now a standard model [29]. A full Bayesian analysis using this model is available on WinBUGS (available free from www.mrc-bsu.cam.ac.uk/bugs).

Key Applications

Disease Mapping

In this area, focus is on the processing of the disease map to take out random noise. Often applications in health services research require the production of an 'accurate' map of relative risks. Models for relative risk range from simple SMRs to posterior expected estimates from Bayesian models. In the count data situation, define the model for the observed counts as

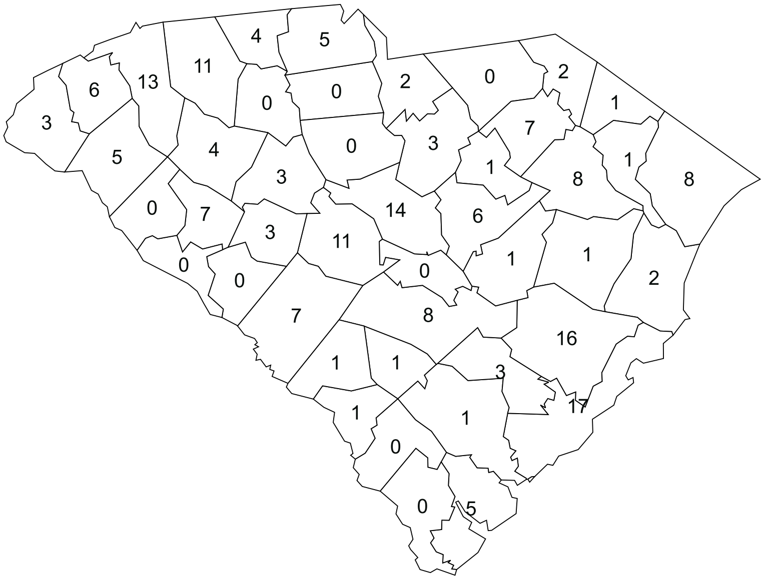
$$\begin{aligned} y_i &\sim \mathrm{Poisson}(e_i\theta_i) \\ \log\theta_i &= \mathbf{x}_i^{\mathrm{T}}\beta + \text{random effect terms}\,, \end{aligned}$$

where $\mathbf{x}_i^{\mathrm{T}}$ is the i th row of a covariate design matrix and **\beta** is a regression parameter vector.

The simplest model assumes no linkages to covariates or random terms and the ML estimator of θ_i is the SMR: i. e. $\widehat{\theta_i} = y_i/e_i$. More often, and more generally, $\log\theta_i$ is assumed to be equal to a linear predictor involving covariates and regression parameters ($\mathbf{x}_i^{\mathrm{T}}\beta$). The final extension includes random effect terms to allow for overdispersion (uncorrelated heterogeneity UH: v_{i}) and spatially-correlated heterogeneity (CH: u_{i}). The model would then take the form:

$$\log\theta_i = \mathbf{x}_i^{\mathrm{T}}\beta + v_i + u_i\,.$$

In applications without covariates, when simple smoothing of rates is required, a simpler random effect model would



Public Health and Spatial Modeling, Figure 3
South Carolina: counts of congenital deaths by county in 1990

be used:

$$\log \theta_i = v_i + u_i \, .$$

This model assigns noise to two components: UH and CH. Both components are usually fitted to capture all the noise components thought to be present. This is often termed the BYM convolution model. To be able to estimate these components, prior distributions are assumed for each component. Usually these consist of an uncorrelated zero mean normal distribution for the overdispersion:

$$v_i \sim N(0, \tau_v) \, ,$$

where τ_v is a variance parameter, and a spatial correlation prior distribution for the CH component. This could be chosen in a variety of ways. Commonly a Markov random field (MRF) is assumed. The intrinsic singular Gaussian distribution ([21,5,30]) is used where the conditional mean of the region effect is based only on a neighborhood of the region:

$$[u_i|......] \propto N(\overline{u}_{\delta_i}, \tau_u/n_{\delta_i})$$

where δ_i is a neighborhood of the i th area, and n_{δ_i} is the number of regions in the i th neighborhood, and τ_u is a variance parameter which controls the degree of smoothing. This can be sampled within a posterior distribution sampling algorithm relatively simply. One alternative to this specification is to assume a fully parametrized covariance and a Multivariate normal distribution for CH:

$$\mathbf{u} \sim \mathbf{N}_p(\mathbf{0}, \Sigma)$$

where the elements of Σ are $\sigma_{ij} = cov(u_i, u_j)$. These covariance elements can be parametrized with a distance-based form such as $\sigma_{ij} = c_0 + \tau \exp(-\alpha d_{ij}^{\psi})$. Here a sill and nugget effect are specified and at zero distance the instantaneous variance is $c_0 + \tau$. This is more heavily parametrized than the MRF model above and the model also requires the inversion of a $p \times p$ covariance matrix. This of course allows for more detailed covariance modeling.

In a full Bayesian analysis, all parameters $(\beta, \mathbf{u}, \mathbf{v}, \tau_* \ldots)$ would be assigned prior distributions and posterior sampling of these parameters, usually via MCMC algorithms, would be required.

For case event data, point process models must be considered initially. A heterogeneous Poisson Process model could be considered for p case events $\{\mathbf{s}_i\}$ $i = 1, \ldots, p$. It is possible to extend such a model to deal with random effects also. However when a control disease is also available, then it is possible to consider a simpler conditional logistic analysis. Define the joint realization of p cases and q controls as $i = 1, \ldots, p$ for the cases and $i = p+1, \ldots, p+q$ for the controls. Assume that the first order intensity of the cases is $\lambda(s, \theta) = \rho\lambda_0(s, \theta)\lambda_1(s, \theta)$ and of the controls $\lambda_0(s, \theta)$. Define the binary indicator variable y_i as follows:

$$y_i = \begin{cases} 1 & \text{if } \mathbf{s}_i \text{ is a case} \\ 0 & \text{otherwise} \end{cases}$$

then the conditional probability of a case at s_i is just

$$\frac{\rho\lambda_1(\mathbf{s}_i, \theta)}{1 + \rho\lambda_1(\mathbf{s}_i, \theta)} \, .$$

Hence, the likelihood of the realization is a logistic likelihood ([8]) specified by

$$L(\theta|\{\mathbf{s}_i\}) = \prod_{l=1}^{p+q} \frac{[\rho\lambda_1(\mathbf{s}_i, \theta)]^{y_i}}{1 + \rho\lambda_1(\mathbf{s}_i, \theta)} . \quad (4)$$

A suitable specification for the relative risk $\lambda_1(\mathbf{s}_i, \theta)$ could be $\log \lambda_1(\mathbf{s}_i, \theta) = \mathbf{x}_i^{\mathrm{T}}\beta + v_i + u_i$ where any covariates would have to be available at all case and control locations. Note that a model without covariates only requires random effect estimates at locations. Specifying suitable prior distributions for such a model is not difficult and, for example, first order neighborhoods of points can be obtained from tessellation information ([2]), and so MRF prior distributions can be specified. Alternative semi-parametric models have been suggested by [17].

Disease Clustering

In this area, the focus is not on reduction of noise, per se, but the assessment of the clustering tendency of the map and in particular the assessment of which areas of a map display clustering. Here, clustering could be around a known putative source of hazard (*focused* clustering) or have no known locations of clustering (*non-focused* clustering). A variety of testing methods are available for cluster detection, see for example [19].

However it is also possible to consider modeling clusters. In general the model formulation may not differ greatly from that of relative risk estimation, depending largely on the definition of clusters and clustering.

Focused Clustering Focused clustering is the simplest case and usually assumes that some form of distance decrease in risk happens around a fixed point or points.

For example, the count data model can be defined as

$$y_i \sim \text{Poisson}(e_i\theta_i)$$
$$\log \theta_i = \log(1 + \exp\{-\alpha d_i\}) + \mathbf{x}_i^{\mathrm{T}}\beta + z_i^{\mathrm{T}}\gamma ,$$

where d_{i} is a distance measured to the small area from the focus point (such as a chimney, mobile phone mast, or waste dump site). Here the extra covariates appear in $\mathbf{x}_i^{\mathrm{T}}$ while the z_i^{T} is the i th row of a matrix of random effects and γ is a unit vector. In this case focus is on inference concerning α as this defined the distance relation. Within $\mathbf{x}_i^{\mathrm{T}}$ there could also be directional terms such as $\cos(\phi)$and $sin(\phi)$,where ϕ is the angle between the area (centroid) and the focus point. This can be used to detect any directional concentration of risk (which could be important particularly if an air pollution risk is possible).

For case events, the case event locations are often assumed to follow a heterogeneous Poisson process with first order intensity $\lambda(\mathbf{s})$. Denote this as $\{\mathbf{s}_i\} \sim PP(\lambda(\mathbf{s}))$.If a control disease is available and the conditional logistic likelihood (4) is assumed then the intensity can be parametrized as:

$$\log \lambda_1(\mathbf{s}_i, \theta) = \log(1 + \exp\{-\alpha d_i\}) + \mathbf{x}_i^{\mathrm{T}}\beta + z_i^{\mathrm{T}}\gamma ,$$

where d_{i} is the distance from any case or control event to the focus point. Directional effects can be included here also as for count data. When fixed effects are included only with no covariates, then a frequentist approach would allow the estimation of α via maximum likelihood. Equally, if a Bayesian approach is assumed then all parameters would have prior distributions and the resulting posterior distribution would usually be sampled. Some general references for this area are [9 chap. 9,25 chap. 7].

Non-Focused Clustering When locations of clusters are unknown then the statistical task becomes more difficult. Not only are the locations of putative clusters unknown but their number and size are also not predefined. This area can be further divided into *general* clustering, where the overall tendency of an area to cluster is assessed, and *specific* clustering where the locations of clusters are to be assessed. Many testing procedures have been derived to assess general clustering tendency (see e. g. for case events: [1,7]; and for counts: [20,34]). Fewer procedures are available for specific clustering. Scan statistics (SatScan) have been proposed ([19]).

Modeling of clusters can be approached in a variety of ways. First, if clustering of excess risk is simply and liberally regarded as *significant excess risk found anywhere on a map* then pointwise determination of excess can be pursued. This is known as hot-spot clustering. For example, for count data, it could be assumed:

$$y_i \sim \text{Poisson}(e_i\theta_i)$$

as before, and examine either *i)* estimates of θ_i for unusual features (usually significantly elevated values), or, *ii)* the residuals from a fitted model:

$$\widehat{r}_i = y_i - e_i\widehat{\theta}_i$$

to find out if, after model fitting, whether there are areas of excess unexplained by the model.

The first approach assumes a model for risk and under that model some form of cluster identification may take place. Alternatively a model which simply cleans noise out may be considered i. e. a model for $\log \theta_i$ is assumed such as $\log \theta_i = \mathbf{x}_i^{\mathrm{T}}\beta + v_i$. This model allows for covariate adjustment and some extra variation but does not model CH (smoothing) as this may reduce its ability to detect aberrations in risk at the single region level. Following the model fit an assessment of the significance of $\widehat{\theta}_i$ could be made.

If on the other hand a specific structure for clusters is assumed then a formal clustering model may be assumed. There is a gray area between relative risk estimation (which focuses on the estimation of θ_i) and i) above where estimates of θ_i are examined for significant excess. If some form of cluster identification is included in the model then that can be checked for location and size of clusters. This can be useful when data are sparse and other global CH models cant describe the cluster form. One proposal is for risk to be related to a set of hidden (unobserved) cluster locations:

$$\log \theta_i = \mathbf{x}_i^{\mathrm{T}} \beta + v_i + \log\{1 + \sum_{k=1}^{K} h(\mathbf{x}_i; \xi_k)\}$$

where there are K unknown clusters with locations $\{\xi_k\}$, $\mathbf{x}_i$ is the centroid of the i th small area and $h(\mathbf{x}_i; \xi_k)$ is a cluster distribution function that describes the relation of any point to a cluster location. Usually $h(\mathbf{x}_i; \xi_k)$ is designed to have a decline in risk with distance from ξ_k, but a range of forms are available. Unfortunately, given that K is unknown, a number of assumptions must be imposed on the analysis to allow for estimation of parameters. Often reversible jump McMC is employed here (see e. g. [13,27,28]). For case event data, models for $\log \lambda_1(\mathbf{s}_i, \theta)$ can be set up with similar considerations (see e. g. [24]).

The second approach, that of examining residuals such as $\widehat{r}_i = y_i - e_i\widehat{\theta}_i$, may be useful if a noise reduction model is used in the estimation of $\widehat{\theta}_i$. However the residual will always include some form of noise unrelated to clustering. Even a perfect model will always have Poisson noise around the true risk: $e_i\theta_i$. Hence it would be important to specify the risk model carefully to allow for only clustering effects to appear in the residual as far as possible. 'Unusual' residuals can be examined via Monte Carlo procedures such as parametric Bootstrap or, under a Bayesian paradigm, a Bayesian Bootstrap using the predictive distribution.

Finally alternative approaches that assume that clusters are defined within areas or neighborhoods (as opposed to single regions) can be considered and diagnostics for these have been proposed [16].

Ecological Analysis

In this area, the relation between disease incidence and explanatory variables is the focus, and this is usually carried out at an aggregate level, such as with counts in small areas.

Many issues of bias and misclassification error can arise with ecological data and the interested reader is referred to [36] and [14] for further insights.

Two important areas of concern are related to scale aggregation issues: MAUP and MIDP. The Modifiable areal Unit Problem (MAUP) concerns the scalability of models and whether, at different spatial scales, a model is valid. In general this is unlikely to be the case as far as covariance structure is concerned as this would lead to fractal covariances which are not found commonly. However, the labeling of scales of relevance of models is important and the extent to which a model can be scaled is relevant in many applications. A related but different issue is how to use different scales of data within one analysis i. e. should individual level data be used in preference to aggregated data or can they be combined. This is a focus of current research.

The misaligned data problem (MIDP) is related to the last issue, but specifically addresses the issue of combining data from different spatial scales to provide analysis at one level. For example health outcomes (disease incidence etc.) may be observed within census tracts and there are available pollution measurements at monitoring sites around the study area. To make inferences about the health outcomes it is best to use the pollution data relevant to the census tracts. One simple solution would be to block Krige the pollution data to provide block estimates for each of the tracts (see e. g. [3, chap. 6,32]). This would ignore the error in the interpolation of the pollution data of course and a better approach is to consider a model where the true exposure is modeled within the health model but the pollution model is jointly estimated.

The model often assumed for count data is of the form

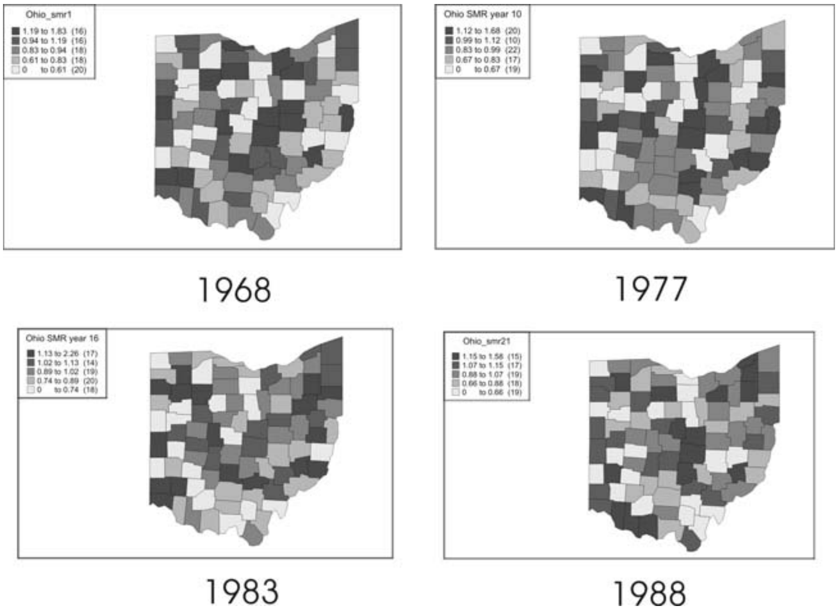
$$\begin{aligned} y_i &\sim \text{Poisson}(e_i\theta_i) \\ \log \theta_i &= \mathbf{x}_i^{\mathrm{T}} \beta + z_i^{\mathrm{T}} \gamma \,. \end{aligned}$$

Assuming that it is possible to observe count data (y_{i}) and also observe measurements $\{x(\mathbf{s}_j)\}$ made at q sites. Assume the measures have mean $E(x(\mathbf{s}_j)) = \mu(\mathbf{s}_j)$, and they are multivariate normal with covariance $\mathrm{cov}(x(\mathbf{s}_i), x(\mathbf{s}_j)) = \sigma_{xij}$. Also Σ is the covariance matrix with ijth element σ_{xij}. For a block, the mean is defined as $\mu_{B_i} = |B_i|^{-1} \int_i \mu(\mathbf{s}_j)\,\mathrm{d}\mathbf{u}$, where B_{i} denotes the ith area, and an estimate is $\widehat{\mu}_{B_i} = |B_i|^{-1} \int_i \widehat{\mu}(\mathbf{s}_j)\,\mathrm{d}\mathbf{u}$. It could be assumed in this case:

$$\begin{aligned} \{y_i\} &\overset{ind}{\sim} \text{Poisson}(e_i\theta_i) \\ \log \theta_i &= \beta \mu_{B_i} + z_i^{\mathrm{T}} \gamma \,. \end{aligned}$$

and jointly with

$$\{x(\mathbf{s}_j)\} \sim N_q(\mu(\mathbf{s}_j), \Sigma)\,,$$



Public Health and Spatial Modeling, Figure 4 Ohio county map: respiratory cancer SMRs for 4 selected years: 1968,1977,1983, and 1988

μ_{B_i} can be estimated and the associated error can be accounted for. Similar considerations can apply to case event data.

Space-Time Modeling and Disease Map Surveillance

Space-Time Models The extension of mapping models to space-time is straightforward in the case of counts within areas within time periods. Figure 4 displays sequences of maps of respiratory cancer for 4 year periods in the counties of the US State of Ohio. Space-time variation in risk is apparent from the variation from year to year for given counties.

For example, yearly counts of disease within small areas can be handled relatively straightforwardly. In this area, the focus is the construction of methods which, usually, examine the spatio-temporal variation of disease. A typical count data model (for counts y_{ij} in the ith region and jth time period) might be

$$y_{ij} \sim \text{Poisson}(e_{ij}\theta_{ij})$$
$$\log \theta_{ij} = \alpha + (\text{covariates}) + u_i^{\mathrm{T}}\gamma + w_j^{\mathrm{T}}\xi + z_{ij}$$

where $u_i^{\mathrm{T}}\gamma$ is a sum of spatial random components γ is a unit vector), and $w_j^{\mathrm{T}}\xi$ is a sum of temporal effects ξ is also a unit vector) and z_{ij} is a space-time interaction effect. This formulation can lead to a rich variety of models depending on the definition of the structure of the components. [18] discusses various possibilities in the Bayesian context.

Map Surveillance Surveillance usually requires there to be a prospective view taken of the data whereby new events are recorded and detection of changes in the vent pattern is important. This may be done in real- or near-real-time. This area has become important due to bioterrorism threats and the possibility of large scale PH disaster prediction. Often a space-time model must be general enough to cope with normal variation in risk but also capable of detecting aberrations as they arise. One useful approach is to consider the predictive distribution of data given previous events and compare this with the new events. This leads to so called *surveillance residuals* [35].

Certain optimal methods are available for the detection of changes in a disease incidence and clustering in space-time (see [11,12,22] provides reviews). General methods for detecting temporal disease changes are given in [10].

Future Directions

There are many open problems in this area. While much attention has been paid to putative hazard assessment (focused clustering) and also methods for relative risk estimation, there is still considerable need for development of methodology for cluster detection and also multi-focus surveillance in real-time. Future directions will see the development of multivariate models and also the fuller examination of space-time.

Cross References

- ► Autocorrelation, Spatial
- ► Biomedical Data Mining, Spatial
- ► Hierarchical Spatial Models
- ► Homeland Security and Spatial Data Mining
- ► Spatial and Geographically Weighted Regression

► Spatial Regression Models
► Statistical Descriptions of Spatial Patterns

Recommended Reading

1. Anderson, N.H., Titterington, D.M.: Some methods for investigating spatial clustering, with epidemiological applications. J. R. Stat. Soc. **160**, 87–105 (1997)
2. Baddeley, A., Turner, R.: Spatstat: An r package for anayzing spatial point patterns. J. Stat. Softw. **12**, 1–42 (2003)
3. Banerjee, S., Carlin, B.P., Gelfand, A.E.: Hierarchical Modeling and Analysis for Spatial Data. Chapman and Hall/CRC Press, London (2004)
4. Bernardinelli, L., Clayton, D.G., Pascutto, C., Montomoli, C., Ghislandi, M., Songini, M.: Bayesian analysis of space-time variation in disease risk. Stat. Med. **14**, 2433–2443 (1995)
5. Besag, J., York, J., Mollié, A.: Bayesian image restoration with two applications in spatial statistics. Ann. Inst. Stat. Math. **43**, 1–59 (1991)
6. Carstairs, V.: Small area analysis and health service research. Community Med. **3**, 131–139 (1981)
7. Diggle, P.J., Chetwynd, A., Haggvist, R., Morris, S.: Second-order analysis of space-time clustering. Stat. Methods Med. Res. **4**, 124–136 (1995)
8. Diggle, P.J.: Point process modellng in environmental epidemiology. In: Barnett, V., Turkman, K.F. (eds.) Statistics in the Environment SPRUCE I. Wiley, New York (1993)
9. Elliott, P., Wakefield, J.C., Best, N.G., Briggs, D.J. (eds.): Spatial Epidemiology: Methods and Applications. Oxford University Press, London (2000)
10. Farrington, P., Andrews, N.: Outbreak detection: Application to infectious disease surveillance. In: Brookmeyer, R., Stroup, D. (eds.): Monitoring the Health of Populations: Statistical Principles and Methods for Public Health Surveillance. chapter 8. Oxford University Press, London (2004)
11. Frisen, M.: Statistical surveillance: optimality and methods. Int. Stat. Rev.. **71**, 1403–1434 (2003)
12. Frisén, M., Sonesson, C.: Optimal surveillance. In: Lawson, A.B., Kleinman, K. (eds.) Spatial and Syndromic Surveillance for Public Health, chapter 3. Wiley, New York (2005)
13. Gangnon, R.: Impact of prior choice on local Bayes factors for cluster detection. Stat. Med. **25**, 883–895 (2006)
14. Gustafson, P.: Measurement Error and Misclassification in Statistics and Epidemiology. Chapman & Hall, London (2004)
15. Haneuse, S., Wakefield, J.: Ecological inference incorporating spatial dependence. In: King, G., Rosen, O., Tanner, M. (eds.) Ecological Inference: New Methodological Strategies, chapter 12, pp 266–301. Cambridge University Press, London (2004)
16. Hossain, M., Lawson, A.B.: Cluster detection diagnostics for small area health data: with reference to evaluation of local likelihood models. Stat. Med. **25**, 771–786 (2006)
17. Kelsall, J., Diggle, P.: Spatial variation in risk of disease: a nonparametric binary regression approach. Appl. Stat. **47**, 559–573 (1998)
18. Knorr-Held, L.: Bayesian modelling of inseparable space-time variation in disease risk. Stat. Med. **19**, 2555–2567 (2000)
19. Kulldorff, M., Nagarwalla, N.: Spatial disease clusters:detection and inference. Stat. Med. **14**, 799–810 (1995)
20. Kulldorff, M., Tango, T., Park, P.J.: Power comparisons for disease clustering tests. Comput. Stat. Data Anal. **42**, 665–684 (2003)
21. Kunsch, H.R.: Intrinsic autoregressions and related models on the two-dimensional lattice. Biometrika. **74**, 517–524 (1987)
22. Lai, T.L.: Sequential changepoint detection in quality control and dynamical systems. J. R. Stat. Soc. B **57**, 613–658 (1995)
23. Lawson, A.B.: On using spatial Gaussian priors to model heterogeneity in environmental epidemiology. The Statistician. **43**, 69–76 (1994). Proceedings of the Practical Bayesian Statistics Conference
24. Lawson, A.B.: Cluster modelling of disease incidence via rjmcmc methods: a comparative evaluation. Stat. Med. **19**, 2361–2376 (2000)
25. Lawson, A.B.: Statistical Methods in Spatial Epidemiology, 2nd edn. Wiley, New York (2006)
26. Lawson, A.B., Böhning, D., Lessafre, E., Biggeri, A., Viel, J.F., Bertollini, R. (eds.) Disease Mapping and Risk Assessment for Public Health. Wiley, Chichester (1999)
27. Lawson, A.B., Clark, A.: Markov chain Monte Carlo methods for clustering in case event and count data in spatial epidemiology. In: Halloran, M.E., Berry, D. (eds.) Statistics and Epidemiology: Environment and Clinical Trials, pp 193–218. Springer, New York (1999)
28. Lawson, A.B., Denison, D.: Spatial cluster modelling: an overview. In: Lawson, A.B., Denison, D. (eds.) Spatial Cluster Modelling, chapter 1, pp. 1–19. CRC press, New York (2002)
29. Mollié, A.: Bayesian and empirical bayes approaches to disease mapping. In: Lawson, A., Biggeri, A., Boehning, D., Lesaffre, E., Viel, J.-F., Bertollini, R. (eds.) Disease Mapping and Risk Assessment for Public Health, chapter 2, pp. 15–29. Wiley, Chichester (1999)
30. Rue, H., Held, L.: Gaussian Markov Random Fields: Theory and Applications. Chapman & Hall/CRC, New York (2005)
31. Salway, R., Wakefield, J.: Sources of bias in ecological studies of non-rare events. Environ. Ecol. Stat. **12**, 321–347 (2005)
32. Schabenberger, O., Gotway, C.: Statistical Methods for Spatial Data Analysis. Chapman & Hall, London (2004)
33. Snow, J.: On the Mode of Communication of Cholera. Churchill Livingstone, London, 2nd edn. (1854)
34. Tango, T.: Comparison of general tests for spatial clustering. In: Lawson, A.B., Biggeri, A., Boehning, D., Lesaffre, E., Viel, J.F., Bertollini, R. (eds.) Disease Mapping and Risk Assessment for Public Health, chapter 8. Wiley, New York (1999)
35. Vidal-Rodeiro, C., Lawson, A.: Monitoring changes in spatio-temporal maps of disease. Biom. J. **48**, 1–18 (2006)
36. Wakefield, J.: A critique of statistical aspects of ecological studies in spatial epidemiology. Environ. Ecol. Stat. **11**, 31–54 (2004)

PVD

► Floating Car Data

Pyramid Technique

CHRISTIAN BÖHM
Institute for Computer Science Database and Information Systems, University of Munich, Munich, Germany

Synonyms

Pyramid tree

Definition

The *Pyramid Technique* [1] is an indexing technique for point data (feature vectors) of a multidimensional space, particularly designed for medium to high dimensionality starting from $d = 10$. Like Z-ordering [2] and other space-filling-curve techniques the pyramid technique gives a one-dimensional embedding of the high dimensional points. The embedded objects can be indexed by any one-dimensional index structure which supports range queries (interval queries) such as all B-tree [3] variants as well as all order preserving hashing methods. The pyramid technique can efficiently handle multidimensional interval queries and nearest neighbor queries using maximum metric.

Historical Background

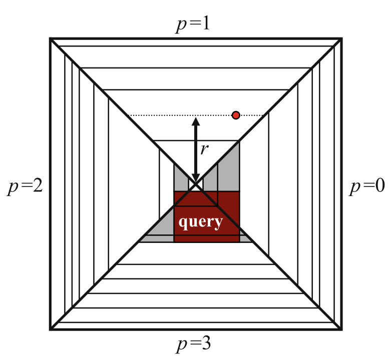
Index structures for vector spaces of medium to high dimensionality [4,5,6] have become very popular in the 1990s, because traditional index structures for vector data, such as the R-tree [7] and its variants tend to deteriorate as the dimensionality of the space increases, an effect commonly referred to as the *curse of dimensionality* [8]. One-dimensional embedding techniques [2], in general, yield the advantage that the complete storage management is handled by an index structure for a one-dimensional space, which is readily available by commercial database systems. Therefore, features such as transaction processing, concurrency and recovery are inherited from the one-dimensional index. Although vector data are transformed into one-dimensional spaces, the complete technique is subject to the *curse of dimensionality*. Therefore, it is important to develop the pyramid technique, which is particularly designed for higher-dimensional spaces and suffers its problems to a lesser extent. The pyramid technique is, in general, not limited to a particular metric, but the schema of space partitioning makes it particularly suited for queries using the maximum metric. The pyramid technique has inspired a number of other techniques, such as the onion technique [9] or concentric hyperspaces [10] and many others, which focus on different metrics including the Euclidean metric.

Scientific Fundamentals

In contrast to most of the well-known space-filling curves, the pyramid technique does not rely on a *recursive* schema of space partitioning. In contrast, the data space is partitioned into $2 \cdot d$ hyper-pyramids which share the origin of the data space (which can be chosen as the center point of the data set) as top point and have each an individual $(d{-}1)$-dimensional basis area (cf. Fig. 1). The pyramids are systematically numbered which forms the first part of the embedding key (a natural number p). The second part is the distance (with respect to the maximum metric) from the origin (a positive real number r). The embedding key can be formed as an ordered pair $k = (p, r)$, or, equivalently, if the maximum of all r-values ($r_{\max}$) is known, we can form one single embedding key $k' = r_{\max} \cdot p + r$.

In both cases, a d-dimensional range query can be translated in a set of search intervals on the search keys. The number of intervals is at most $2 \cdot d$, because the query object can at most have one intersection with each of the pyramids (cf. Fig. 1. Since nearest neighbor queries can be transformed into range queries (which requires a set of at most two one-dimensional ranking processes per pyramid), it is also possible to evaluate nearest neighbor queries.

The *Extended Pyramid-Technique* was proposed to handle data with skewed data distribution. The idea is a translation of the the data set such that the center of the set is located at the reference point $(0.5, \ldots, 0.5)$ where all the pyramids of the original pyramid techniques share their top point while keeping the data in the unit hypercube $[0..1]^d$. Since the centroid (means) of the data points is not very stable in the presence of skewed data distributions, our technique is based on a median method which determines a point, which is the coordinate-wise median of all data points. We refer to this point as the median point $mp = (mp_1, \ldots, mp_d)$. Then we determine for each coordinate an expo-



Pyramid Technique, Figure 1 Space partitioning of the pyramid technique

P

nent r_i such that the following condition holds:

$$r_i = -\frac{1}{\log_2(mp_i)}\,.$$

This exponent is used in a function $t(x) = (x_1^{r_1}, \ldots, x_d^{r_d})$ to transform the data in a new space from which it has the nice property to map the points $[0..1]^d$ into the same space $[0..1]^d$, while moving the median point mp to the reference point $(0.5, \ldots, 0.5)$ of the pyramids. The index must be rebuilt whenever the change of the median point mp extends a threshold. It was shown that this happens rarely. In addition to the general advantages of the one-dimensional embedding, the experimental evaluation of the pyramid technique and the extended pyramid technique yielded a considerable speed-up factor of up to 14 with respect to the number of page accesses, of up to 103 with respect to CPU consumption and of up to 2500 with respect to the overall response time over the X-tree.

Key Applications

High dimensional indexing is important for similarity search systems in various application areas, such as multimedia, CAD, systems biology, medical image analysis, time sequence analysis and many others. Complex objects are typically transformed into vectors of a high-dimensional space (feature vectors), and the similarity search thereby translates into a range or nearest neighbor query on the feature vectors. High-dimensional feature vectors are also required for more advanced data analysis tasks such as cluster analysis or classification.

Future Directions

One-dimensional embedding would also be interesting for several new metrics, such as set metrics (multi-instance objects) or for uncertain and moving objects. Only few approaches exist [11] to support general metric spaces.

Recommended Reading

1. Berchtold, S., Böhm, C., Kriegel, H-P.: The pyramid-technique: towards breaking the curse of dimensionality, pp. 142–153, SIGMOD Conference, (1998)
2. Orenstein, J.A.: Redundancy in spatial databases. SIGMOD Conference pp. 295–305 (1989)
3. Bayer, R., McCreight, E.M.: Organization and maintenance of large ordered indexes, pp. 107–141, SIGFIDET Workshop (1970)
4. Lin, K-I., Jagadish, H.V., Faloutsos, C.: The TV-tree: An index structure for high-dimensional data. VLDB J. 3(4):517–542 (1994) [DBLP:journals/vldb/LinJF94]
5. White, D., Jain, R.,: Similarity indexing with the SS-tree. ICDE 516–523 (1996)
6. Berchtold, S., Keim, D.A., Kriegel, H-P.: The X-tree: An index structure for high-dimensional data. VLDB, 28–39 (1996)
7. Guttman, Aa.: R-Trees: A dynamic index structure for spatial searching. SIGMOD Conference 47–57 (1984)
8. Berchtold, S., Böhm, C., Keim, D.A., Kriegel, H-P.: A cost model for nearest beighbor search in high-dimensional data space. PODS 78–86 (1997)
9. Chang, Y.-C., Bergman, L.D., Castelli, V., Li, C.-S., Lo, M.-L., Smith, J.R.: The onion technique: indexing for linear optimization queries. SIGMOD Conference 391–402 (2000)
10. Ferhatosmanoglu, H., Agrawal, D., Abbadi, A.E.: Concentric hyperspaces and disk allocation for fast parallel range searching. ICDE 608–615 (1999)
11. Jagadish, H.V., Ooi, B.C., Tan, K.-L., Cui, Yu, Rui, Zhang: iDistance: An adaptive B+-tree based indexing method for nearest neighbor search. ACM Trans. Database Syst. 30(2):364–397 (2005)

Pyramid Tree

► Pyramid Technique

QGIS

► Quantum GIS

Qt Libray

► Quantum GIS

Q-Tree

► Quadtree and Octree

Quadtree

► Quadtree and Octree

Quadtree and Octree

MARCI SPERBER
Department of Computer Science and Engineering,
University of Minnesota, Minneapolis, MN, USA

Synonyms

Quadtree; Q-tree; Octree; Data-structure, Spatial; Point-Quadtree; MX-Quadtree; PR-Quadtree; PM-Quadtree

Definition

A quadtree is a spatial data structure which has four branches attached to the branch point or node. The records exist in the leaf nodes of the tree. An octree is the same concept except the branches are in groups of eight. An octree can represent and image by subdividing the cubical volume. The quadtree tree is greatly used for two-dimensional space and the octree is used for three-dimensional space.

Historical Background

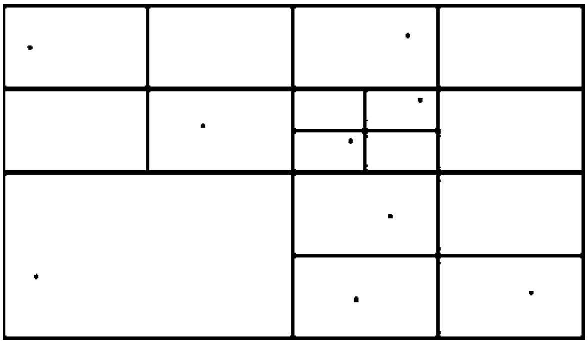
The name quadtree has developed through time. It was once called a Q-tree and then later termed quadtree. It was adapted from the binary search tree in order to be used for two dimensions. The name octree simply comes from the prefix "oct" and the word tree. These data structures were needed in order to save space. These structures were first built as pointers, but have now evolved to use leaf nodes encoded by a locational code.

Scientific Fundamentals

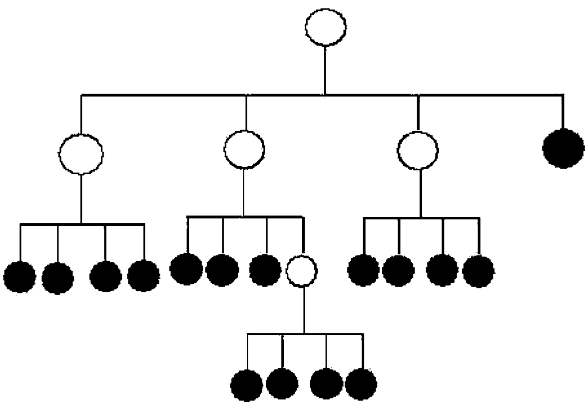
A quadtree is a class of spatial data structure methods used to put and/or locate files or records in a database. There is a base node which is split along d dimensions producing 2^dchildren. Quadtrees are based on the rule of four, and therefore the order of the tree is four and there are four branches attached to the branch point or node. The terminal branches of the tree are referred to as leaf nodes since they are the end of the tree, and the records exist in those leaf nodes. A leaf node that does not contain a record is considered (to be) null. The data in the database is found by repeatedly dividing the number of files or records into four parts until there is only one left.

An octree data structure consists of a cell with eight children where the cell is a cube in space. Octrees are the three-dimensional version of quadtrees. The octree is used to index tree dimensions by subdividing a region of space into eight equal partitions called cells. Cells are divided using specific conditions like containing an object boundary or containing a specific number of objects.

The octree data structure represents objects in three-dimensional space. An octree is a spatial decomposition where the root of the tree is recursively divided by two in each coordinate direction until each cell holds at maximum a specified number of objects. The octree groups its objects hierarchically, which avoids the representation of empty portions of the space. The first node of an octree is called the root node or cell. The root node or cell points to eight elements or cells in which each element or cell can also point to eight children elements or cells, and so on.



Quadtree and Octree, Figure 1 Example of a quadtree representation of random points in a plane

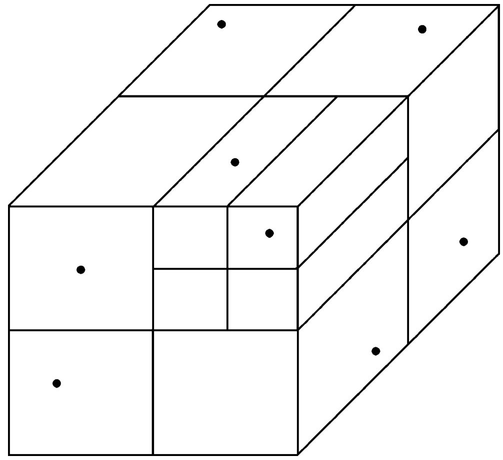


Quadtree and Octree, Figure 2 Example of the quadtree representation with 16 leaves

The last level reached is called the leaf level, which holds the leaf elements or cells. If every cell above the leaf level points to a cell, then the octree is considered a full octree. Figures 1–4 illustrate examples of quadtrees and octrees using points.

Key Applications

Quadtrees are used in a number of situations including measurement of area, finding adjacent leaves, image representation, spatial indexing, collision detection in two dimensions, and measuring the area of contiguous patches. A quadtree is a very useful algorithm to locate pixels in a two-dimensional image because square pixels can be divided into four square parts over and over again. The depth of such a tree depends on image resolution, computer memory and image complexity. Quadtrees are also commonly used to perform search and retrieval on point-based data.



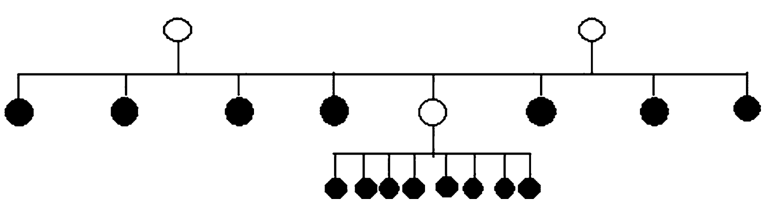
Quadtree and Octree, Figure 3 Example of an octree representation of points in three-dimensional space

Octrees, instead of a method that uniformly divides space, can make the execution of queries faster and save on memory. Octrees can assist with modeling in many areas of earth science (such as geographical terrain, mineral deposits, and geological data), industrial modeling, robotics, pattern recognition, computer vision, and even medical imaging. Specifically, octrees can be used for spatial indexing, collision detection and view frustum culling.

Storing and Accessing Quadtrees

There are different approaches to storing quadtrees: The value of either the block or the pointer can be stored in each memory location. The stored pointer would point to the first four child blocks at the next level down in the tree. The four child blocks of the parent will occur together.

There are also different ways of accessing data from a quadtree: all parts of a map with a given value can be found or the contents of a given pixel can be determined. If the array has $2n \times 2n$ pixels, then there would be n possible levels in the tree, not counting the top level or node, which is considered level zero. To find all the parts of a map with a given value, each leaf is examined to see if a value matches the one required. This would be m steps if there were



Quadtree and Octree, Figure 4 Example of the octree node representation which contains 16 leaves

m leaves in the tree. To determine the contents of a given pixel, follow the branch of the tree containing the pixel. To find out which branch to follow, write the row and column numbers in binary, interleave the bits and convert the number to base 4. The appropriate digit is used at each level to find the branch to follow. This may take n steps in the worst case scenario since it may have to go to level n of the tree to find the contents of the pixels. This only has to be done if the tree is not homogeneous since, if it is homogeneous, the contents are already known.

To transverse a quadtree, simply move down the leftmost branch of the first leaf. After each leaf in the branch is processed, move back up to the branching point and move right. This will move you to the next branch or back to the previous branching point.

Representing Octrees

Octrees can be represented in a hierarchical tree structure or a pointerless linear tree structure. A hierarchical tree structure's cell list includes the root cell, intermediate cells and leaf cells. Each cell includes pointers to parent and child cells. The point location and neighbor search operations would follow pointers to traverse the tree. A linear tree structure's cell list includes only leaf cells. Each cell has a locational code which is used as a search key for cell location. Locational codes are composed of the cells' minimum coordinates. A linear tree is usually more compact but can be more costly and complex when considering processing methods.

To manage the information held in octrees, basic tree traversal techniques including point location, region location and neighbor searches are used. Point location locates a leaf cell which holds the point being searched for. Point location requires moving down through the tree and comparing the position of the point to the position of the current cell. In a linear tree, the position of the point would be converted to the locational code, and the search would be performed by finding the cell that best matches the locational code. Neighbor search locates a cell in a specified direction which touches a given cell. This search requires moving upward recursively through the tree to find the smallest ancestor of both the cell and its neighbor, and then moving downward to find its neighbor.

Point location, region location, and neighbor search often require costly comparisons, since their predictive branching strategies can cause problems when the branching is not predicted correctly. The recursive nature of neighbor search has a lot of overhead, thereby adding to its cost.

Types of Quadtrees and Octrees

The five main types of quadtrees are: point quadtree, matrix (MX) quadtree, point region (PR) quadtree, Bucket point region (PR) quadtree, point matrix (PM) quadtree, and point matrix region (PMR) quadtree.

Point quadtrees are a combination of uniform grid and binary search trees. Introduced by Finkel and Bentley [8], they are useful for spatial data structures, indexing data structures, and low-dimensional points. They seem to be the simplest quadtrees to understand but are usually not the simplest to implement. Point quadtrees are a lot like binary trees, but they use x and y coordinates instead of data to decide where to put the data, which is where the name point quadtree came from. Unlike binary trees, which have nodes with only two children, point quadtrees contain nodes with four children. The four children of the point quadtree are usually called northeast, southeast, southwest, and northwest. To construct a point quadtree, place the first point at the root and then split the underlying space into 2^d smaller regions or quadrants. Each of these regions corresponds with one of the 2^d subtrees of the root. The remaining points are directed to the quadrants and subtrees are constructed recursively. The nodes are inserted by finding the quadrant in which a point belong to relative to the root of the subtree. Thus if the current node is (0,0) and you want (1,1), you would go to the northeast child. When an empty node is reached, insert the point. So just like a binary tree, the structure of the point quadtree is dependent on the values which are inserted into it. Although algorithms have been developed for deleting nodes, they are not considered ideal. Point quadtrees work especially well for performing proximity searches because they work well at finding points that are closer to one point than another. They are great at finding points that are closer to one point than another.

MX Quadtrees, leaf nodes indicate an absence or presence of a data point in the appropriate position in the matrix. A region split will occur in the middle of a region. An MX quadtree decomposes space into regions in an absolute manner. Specifically, it divides a fixed space into one by one square regions. It is not used to represent unbounded spaces. The points that the MX quadtree contains are bounded by a square with sides of length 2^N, where N is an integer. To represent a region that is not square, create a MX quadtree big enough to bound the whole region. To make things simpler, change the resolution so the smallest space between two points corresponds to a unit value. Only leaf nodes can store data in an MX quadtree, unlike in a point quadtree. The internal nodes will get us only to the quadrants. All the data is contained in the leaf nodes in level n of the tree, where the first node is level zero. Empty space in an MX quadtree is merged into empty leaf nodes at lower levels, making these trees very memory efficient for representing sparse matrices. An MX quadtree works well for range search. It is a good choice for such things as determining locations of all bus stops within a five-block radius of a location.

PR quadtrees represent a regular decomposition point, representation meaning they decompose whenever a block contains more than one point (described in more detail below). A PR quadtree is different from an MX quadtree because it subdivides space into quadrants based on the points it contains. The points are inserted into the tree similarly to the point quadtree, but the data points are not stored in internal nodes, much like the MX quadtree. If you reach a nonempty leaf node when inserting a point, the space containing the node needs to be subdivided until only one point is contained in each quadrant. This splitting of the nodes makes inserting more complicated than inserting into a point or MX quadtree.

The Bucket PR quadtree decomposes the underlying space recursively into four equal area blocks. This is done until the number of data points in each block fits in the bucket capacity or size. This means the points of partition are independent of the data. The bucket PR quadtree was an adaptation of the PR quadtree in order to work with data stored in disk pages. The bucket size is usually equal to the number of data records that fit in a single disk page.

The PM quadtree uses the regular decomposition of space. It has been found to be useful for line data and gets its name from representing polygonal maps. PM quadtrees can be built by dividing a collection of vertices and edges into a simple subset. The leaves of the quadtree may contain only one vertex and the edges incident to that vertex.

The PMR quadtree is especially good for spatial indexing of queries that have spatial joins. This is an edge-based data structure where line segments are stored. These line segments are stored based on all the blocks that the line intersects. The PMR quadtree uses the probabilistic splitting rule. If the size of the bucket is exceeded, then it is split into four equal quadrants.

There are also different types of octrees, including PR octrees and MX octrees, which are three-dimensional versions of the quadtrees described earlier.

Future Directions

Future directions dealing with quadtrees and octrees could include the development of new storage techniques and new uses.

Cross References

► Data Models in Commercial GIS Systems
► Indexing, Hilbert R-tree, Spatial Indexing, Multimedia Indexing

Recommended Reading

1. Walid G.A., Ilyas, I.F.: A Framework for Supporting the Class of Space Partitioning Trees. TR01-002. Purdue University, West Lafayette, IN (2001)
2. Black, P.E. Quadtree. In: Dictionary of Algorithms and Data Structures. NIST Available via: http://www.nist.gov/dads/HTML/quadtree.html (2006)
3. Chandran, S., Gupta, A.K., Patgawkar, A.: A Fast Algorithm to Display Octrees. (2004) http://www.cse.iitb.ac.in/~sharat/papers/linearOctreee.pdf
4. Charlton, E.F.: The Octree. (1997)http://hpcc.engin.umich.edu/CFD/users/charlton/Thesis/html/node29.html. University of Michigan, Ann Arbor, MI, USA
5. Chern, H.-H., Fuchs, M., Hwang, H.-K.: Phase Changes in Random Point Quadtrees. National Taiwan Ocean University, National Chiao Tung University, and Institute of Statistical Science Academia Sinica, Taiwan. ACM Transact Algo 3(12):2 (2007)
6. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. MIT Press. Cambridge, MA (2001)
7. Finkel, R.A., Bentley, J.L.: Quad trees, A data structure for retrieval on composite keys. Acta Informatica (1974)
8. Frisken, S., Perry, R.: Simple and Efficient Traversal Methods for Quadtrees and Octrees. http://www.merl.com/reports/docs/TR2002-41.pdf
9. Heckbert, P.S., Garland, M.: Survey of Polygonal Surface Simplification Algorithms. In: SIGGRAPH 97 Course Notes, No. 25, ACM Press, New York (1997)
10. Hjaltason, G.R., Samet, H., Sussmann, Y.J.: Speeding up Bulk-Loading of Quadtrees. In: Proceedings of the 5th ACM international workshop on Advances in Geographic Information Systems, University of Maryland, Las Vegas, Nevada, USA (1997)
11. Meagher, D.J: Octree 3-D Geospatial Modeling Application: Artillery Shadows. http://www.octree.com/ArtilleryShadow/report.htm. Accessed 12 March 2007
12. Samet, H.: Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS. Addison-Wesley, Reading, MA (1990)
13. Savarese, D.F.: A Trio of Quadtrees. JAVAPro 7 November (2003)
14. Tayeb, J.: Design and Performance Evaluation of Indexing Methods for Dynamic Attributes in Mobile Database Management Systems. Master's thesis, Bilkent University, May 1997, Ankara, Turkey (1997)
15. Yoder, R.C.: Design Consideration for Implementing Complex Client/Server Applications: a Case Study of Octree-Based 3-D GIS. (1996)http://www.albany.edu/~sysrcy/dissertation/PR-11.TXT. Accessed 8 April 2006

Qualitative Similarity

► Conceptual Neighborhood

Qualitative Spatial Reasoning

► Geosensor Networks, Qualitative Monitoring of Dynamic Fields

Qualitative Spatial Representations

► Geosensor Networks, Qualitative Monitoring of Dynamic Fields

Quality of Information

► Computing Fitness of Use of Geospatial Datasets

Quality of Services

► Network GIS Performance

Quality, Spatial Data

► Imprecision and Spatial Uncertainty

Quantization

► Indexing, High Dimensional

Quantum GIS

MARCO HUGENTOBLER
Institute of Cartography, ETH Zurich,
Zurich, Switzerland

Synonyms

QGIS; Open Source; Qt libray; MapServer

Definition

Quantum GIS [1] is an opensource GIS written in C++ available under the conditions of the GPL license. It is based on the C++ crossplattform library Qt from Trolltech [4]. Therefore, it runs on most existing operating systems, including Linux, Unix, Mac OS X and Windows. The name 'Quantum GIS' has no special meaning, except that it starts with a Q, indicating that it uses the Qt library. The main focus of QGIS is interactive two dimensional viewing of spatial data. However, there is also functionality for editing vector data and a GRASS plugin to use the analytical functionality of the GRASS program [3] from within the QGIS GUI. QGIS supports a large number of vector and raster formats, including PostGIS, GRASS, Shapefile, GML, WFS, GPX, WMS, GeoTiff, PNG, JPG and many others. QGIS supports reprojecting on-the-fly for vector data sets by using the PROJ4 library.

QGIS provides a plugin mechanism which can be used to add support for a new data source or to extend the functionality of the main program in a modular way.
Figure 1 shows a screenshot of the QGIS program with the map window, the legend and the attribute table opened.

Historical Background

The QGIS project was founded by Gary Sherman in 2002. He was looking for a fast geographic data viewer that runs on Linux and supports a broad range of data sources. However at this time, most commercial software vendors in the GIS domain sold their desktop products only for one operating system. Therefore, he decided to start a new software project. In June 2002, QGIS was officialy registered as a project on the SourceForge webpage. The first CVS checkin dates were from July 6th 2002. The QGIS project continuously attracted more software developers and users. Today (December 2006), QGIS consists of approximately 94000 lines of code and is used by thousands of people all around the world.

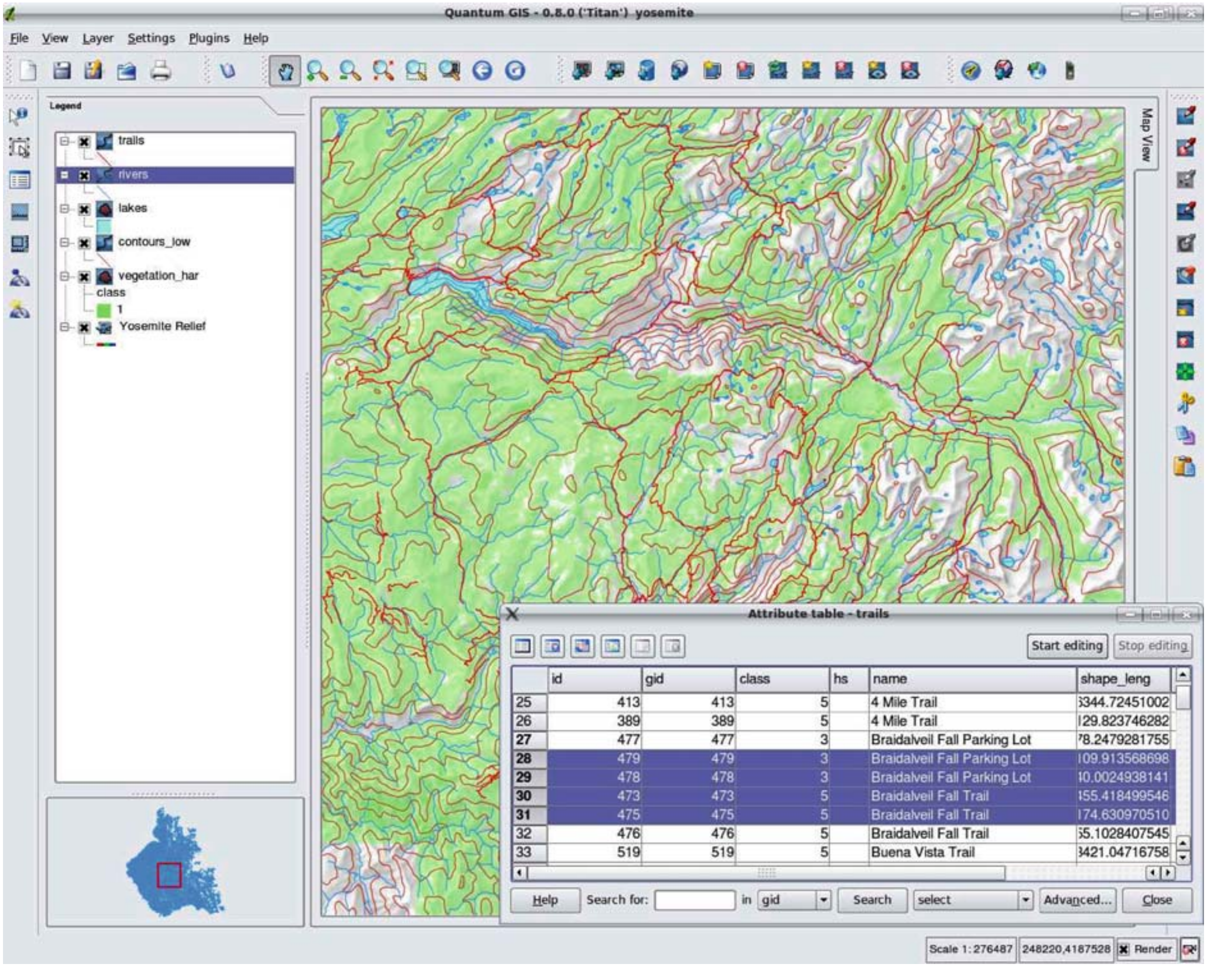
Scientific Fundamentals

Features

In order to have an idea of the functionality provided by QGIS, the main features are summarized below. Please keep in mind that the number of features constantly increases.

- **Map navigation** The usual set of zoom-, pan-, and info tools are available. Additionally, QGIS provides an overview window to facilitate navigation in the map. To store view extents, QGIS has the mechanism of spatial bookmarks.
- **Attribute table** For vector layers, the thematic attributes can be viewed in a table. Besides showing attribute values, the table contains the functionalities of sorting rows, selecting features, editing values and searching features by attribute values.
- **Vector symbolisation** The outline and fill of vector features can be changed. Features can be grouped into classes (graduated symbol), have continuously varying color (continuous color ramps) based on attributes or have a class for each attribute value (unique value). Each class may have a label which is displayed in the legend. QGIS supports transparency and provides several fill patterns and line styles.
- **Point symbols** QGIS provides a number of predefined point symbols. For custom symbols, it is possible to insert any svg file into the symbol directory and QGIS will recognize it.

Q



Quantum GIS, Figure 1 The user interface of the QGIS program

- **Labeling** Vector features can be labeled by attribute. At the moment, the labels are placed at the points of gravity of the features, which is not always a good choice. Many options are available to choose the appearance of the labels. Some of those options (e. g., the size) may be dependent on attribute values.
- **Map editing** QGIS provides digitizing of vector features in the map window. New Point/Line/Polygon features can be added to or removed from existing data sets. Vertices of existing features can be moved, added or deleted using a snapping tolerance. There is also a mechanism for cut/copy/paste features between different layers. The availability of digitizing features varies depending on data sources.
- **Projections** QGIS provides on-the-fly projection for vector layers. This means that the data source is not changed, but the vertices of the layer are mapped to different locations on screen. Using this reprojection mechanism, it is possible to use maps with data sources in different coordinate systems. At the moment, reprojection is not available for raster layers.
- **Tree view legend** The legend is implemented as a tree view and is highly configurable. Each layer has an entry showing name and symbology of the layer. Layer visibility can be toggeled with a checkbox. Several layers can be grouped together in folders, e. g., to toggle visibility of all involved layers with a single mouse click.
- **Print composer** This feature allows for the creation of cartographic output by generating layouts with map, legend, north arrow and scale bar. Layouts can be sent directly to a printer or (as post script) to files.
- **Project file** All the settings in a session can be stored in a project file. QGIS does not use a native format to store the data sources on disk. Instead it uses the external data formats and stores references to them in the project file.
- **Raster pyramids/spatial index** On some data sources, QGIS may trigger the creation of additional data structures to enhance efficiency. This is possible with pyra-

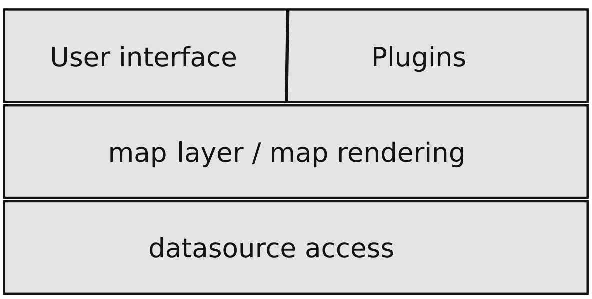
mids for GDAL rasters and with spatial indexes for shapefiles.

- **Scale dependent rendering** Map layers can be visible depending on the current map scale. This is useful for data sets available at different generalisation levels.

General Architecture

The code of QGIS evolves very fast. Nevertheless, this section tries to explain some fundamental design principles which are not likely to be changed in the near future.

The general structure of QGIS can be seen as a set of abstraction layers (Fig. 2). The lowest layer is the data source access. For vector data sources, this layer has the responsibility to read native features from the data sources and convert them to instances of the QGIS feature class (QgsFeature). For raster data sources, this layer hides the complexity of the individual data sources by passing images to the map layers.



Quantum GIS, Figure 2 QGIS code organisation

The second layer is responsible for the organisation into map layers and the rendering of the map. On the top of the layer stack is the graphical user interface. This layer translates the user commands to the map layer and rendering logic. QGIS plugins also interact with the user and therefore are on the same level as the user interface layer. Nevertheless, they may also add a considerable amount of non-GUI functionality.

QGIS Plugins

QGIS follows a modular approach by providing a plugin mechanism to extend the functionality of the main program. There is a plugin interface class which provides access to the QGIS main window. Like this, plugins may add and remove buttons and menus, and may react to mouse events on the map window. Each plugin is compiled into its own library and provides a set of extern C functions such that the QGIS main program has information about it and is able to load it.

Data Providers

From a technical point of view, the data providers which enable the access to individual data sources can be seen as plugins too. They are also loaded to the main program at runtime and are located in the same directory as the previously mentioned plugins. The main difference to the user interface plugins is that data providers are loaded indirectly by the QGIS code and not directly on user demand.

Normally, a data provider library is linked to a second library for native access to a data source. For vector formats, the native library extracts vector features in native format and the QGIS provider converts them to instances of the QGIS feature class (QgsFeature). A QgsFeature contains the thematic attributes as strings and the geometry in WKB (well known binary) format. WKB is a compact format for the storage of simple feature geometries which allows for a fast parsing by the QGIS rendering code. Figure 3 shows the interactions of the different libraries in case of a Postgres database access. QGIS sequentially retrieves the features with calls to the provider function 'getNextFeature()'. The Postgres provider uses libpq for access to the Postgres database and fetches the next row by using a binary cursor. The data structure returned by libpq is then transformed into a QgsFeature and returned to the qgis core.

Key Applications

Visualisation of Vector and Raster Data

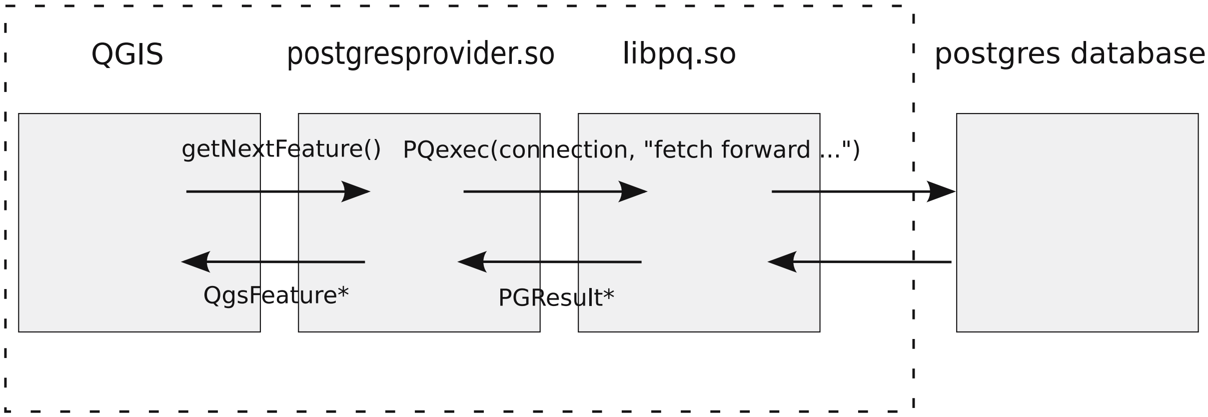
Interactive viewing of spatial data is straightforward. Data layers can be imported and viewed together with other data sets or background raster maps. Zoom- and pan tools as well as an overview window are available for navigation in the map. Although QGIS offers much more functionality for advanced users, this is probably the most frequent use of QGIS.

Import/Export of GPS Data

QGIS supports the input of GPS data with the GPX provider. It is thus quite common that people transfer their routes, waypoints and tracks from their GPS devices to the PC and visualize them on a map, possibly in combination with geodata from other sources. For access to the GPS devices and the conversion into the GPX format, the program GPSBabel is used.

WMS/WFS Client

QGIS can act as a client to servers which offer web map services (WMS) or web feature services (WFS). A WMS data source is handled as a raster layer and QGIS makes



Quantum GIS, Figure 3 Database access through the vector provider interface

a request for each zoom or pan action. WFS is handled as a vector data source. Because data sets can be quite large, QGIS keeps the data in virtual memory after the first request. This makes the zoom and pan actions for WFS layers very fast.

UMN Mapserver Export

A mapserver export tool is part of QGIS. This tool makes the administration of UMN mapserver convenient because it allows one to assemble a map in QGIS and export the QGIS project file to an UMN mapfile.

Spatial Analysis with GRASS and QGIS as a Viewer

In the current version 0.8, QGIS itself has only limited possibilities for spatial analysis. But for spatial analysis, an excellent open source program exists: GRASS. GRASS is, in a way, the opposite of QGIS. It contains a broad range of analysis methods for raster and vector data, but only limited possibilities for interactive viewing. It is therefore a good idea to combine the two programs, using each one for its individual strengths. With the GRASS plugin, QGIS can be used as a spatial data viewer for GRASS vectors and rasters. The integration of the two programs extends even further with the so-called GRASS toolbox. With this toolbox, spatial analysis in GRASS can be started from within the QGIS GUI. Although the analysis is performed by the GRASS program, the user is not exposed to having to change between two different programs.

Third Party Applications with QGIS Libraries

At the beginning of the project, QGIS was just one big executable file. Later, it was split into several libraries to allow third party applications to link with it and reuse the QGIS code. Note that, like QGIS, these libraries are also under the GPL license and so any third party applications linking to them need to be GPL too. An example of such an application is the QGIS WMS server [5], which uses the functionality in the QGIS libraries as a GIS backend and rendering engine. When a request arrives, a map is rendered in memory and sent back over the network.

Future Directions

The future direction of an open source program is always hard to predict because individual developers are able to incorporate their creativity and their own ideas. Some of the main challenges for the future may include the following:

- Many interfaces and data structures were generated some time ago and need redesign in order to provide the best API for version 1.0.
- QGIS needs extensions to do its own spatial analysis. At present, analysis is only possible by using the GRASS software and therefore only for GRASS layers in QGIS.
- One of the current trends in GIS is the increasing importance of distributed systems and web services. QGIS may respond to this by improving client support for the OGC web services or by providing a server module.

Cross References

► GRASS
► Open-Source GIS Libraries
► PostGIS
► Web Feature Service (WFS)

Recommended Reading

1. QGIS (2006) Homepage of the QGIS project. http://qgis.org. Accessed 21 Dec 2006
2. Sherman et al. (2006) QGIS user guide. http://qgis.org/releases/userguide.pdf. Accessed 21 Dec 2006

3. GRASS (2006) Homepage of the GRASS project. http://www.grass.itc.it/. Accessed 21 Dec 2006
4. Trolltech (2006) Qt documentation. http://doc.trolltech.com/. Accessed 21 Dec 2006
5. QGIS WMS (2006) QGIS WMS server homepage. http://karlinapp.ethz.ch/qgis_wms/qgis_wms.html. Accessed 21 Dec 2006

Queries in Spatio-temporal Databases, Time Parameterized

YUFEI TAO
Department of Computer Science and Engineering,
Chinese University of Hong Kong, Hong Kong, China

Synonyms

Continuous queries; Spatio-temporal query; Influence time; Streams

Definition

The output of a *time parameterized* (TP) query has the general form <**R**, **T**, **C**>, where **R** is the set of objects satisfying the query now, **T** is the expiry time of **R**, and **C** is the set of objects that will affect **R** at **T**. From the set of objects in the current result **R**, and the set of objects **C** that will cause changes, we can incrementally compute the next result. We refer to **R** as the *conventional*, and (**T**, **C**) as the *time-parameterized* component of the query.

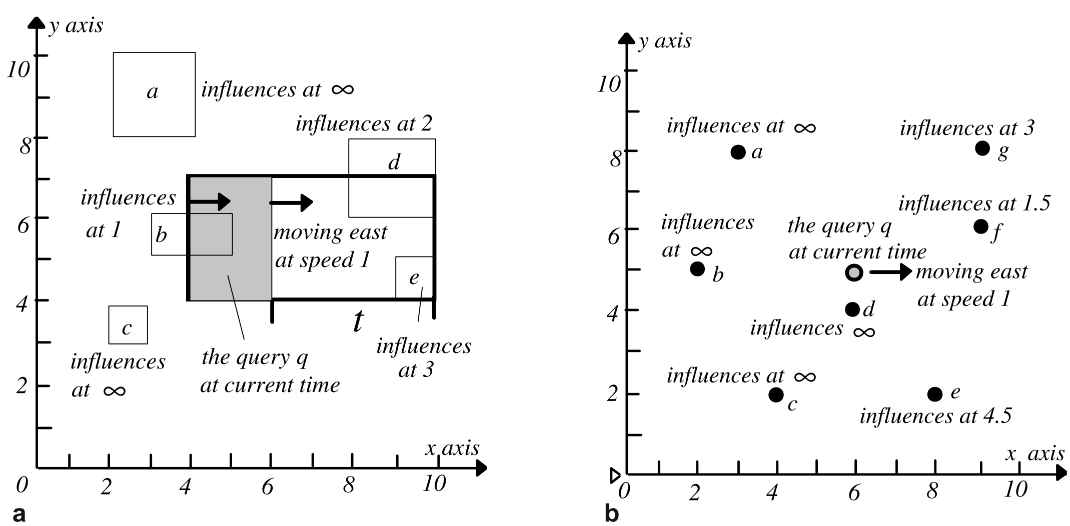
Consider, for instance, the *TP window query* (shaded region) of Fig. 1a, where objects (rectangles *a* to *e*) are static[1] and query *q* is moving east with speed 1. The output is <{*b*}, 1, {−*b*}> meaning that object *b* currently intersects the query window, but after 1 time unit, it will stop doing so (therefore, *b* should be removed from the result which will become empty). Figure 1b demonstrates an example of a *TP nearest neighbor search*. Here, *a*, *b*, . . . , *g* denote data points, and *q* represents a query moving at the direction and speed indicated. The result is <{*d*}, 1.5, {*f*}>, indicating that *d* is now the closest point of *q*, but after 1.5 time units, the nearest neighbor of *q* will become *f*.

Historical Background

Traditional operations (e. g., window search, nearest neighbor retrieval, spatial join, etc.) in spatial databases are "instantaneous" since they are evaluated once and return a single result. Continuous queries, on the other hand, require constant evaluation and updates of the results as the query conditions or database contents change. Such queries are especially relevant to the spatiotemporal databases which are inherently dynamic. Additionally, the result of any query is strongly related to the temporal context.

Any traditional spatial query has a continuous counterpart whose termination clause depends on the user or application needs. Consider, for instance, a window query where the window (and possibly the database objects) moves/changes with time. The termination clause may be temporal (for the next 5 minutes), a condition on the result

[1] For simplicity, static 2D objects are often used. The extension to mobile objects and higher dimensions, unless explicitly stated, is straightforward.



Queries in Spatio-temporal Databases, Time Parameterized, Figure 1 Examples of time-parameterized queries. **a** TP window search, **b** TP nearest neighbor search

(e. g., until exactly one object appears in the query window or until the result changes three times), a condition on the query window (until the window reaches a certain point in space) etc. A major difference from continuous queries in traditional databases [2] is that, in the case of spatiotemporal databases, the object's dynamic behavior does not necessarily require updates, but can be stored as a function of time using appropriate indexes [7]. Furthermore, even if objects are static, the results may change due to the dynamic nature of the query itself (i. e., moving query window), which can also be represented as a function of time. Thus, a spatiotemporal continuous query can be evaluated instantly (i. e., at the current time) using time-parameterized information about the dynamic behavior of the query and database objects in order to produce several results, each covering a validity period in the future.

The concept of a time parameterized search, proposed in [10], is motivated by earlier work on spatiotemporal continuous queries. In [9], the authors describe several modeling and query languages, but do not propose access or processing methods. Song and Roussopoulos [8] processed moving nearest neighbor queries in R-trees by employing sampling. That is, they incrementally computed the results at pre-determined positions, using previous results to avoid total re-computation. This approach is, however, limited in scope (only applicable to nearest neighbors, and static objects). Furthermore, it suffers from the usual drawbacks of sampling, i. e., if the sampling rate is low, the results will be incorrect, or otherwise there is significant computational overhead; in any case, there is no accuracy guarantee since even a high sampling rate may miss some results. Zheng and Lee [12] discuss a restricted version of the problem (moving query, static objects indexed by R-trees) for a single nearest neighbor, using Voronoi diagrams. In addition to the NN of the query point, they return the valid period of the result, which is a conservative approximation obtained by assuming that the query can have a maximum speed. Neither approach can deal with dynamic objects or other types of queries. Recently, Mokbel et al. [4] extended the notion of continuous queries to spatiotemporal streams, and developed efficient processing algorithms.

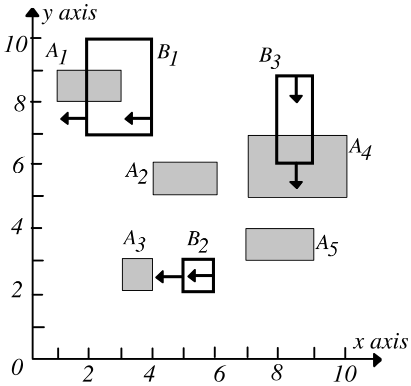
Scientific Fundamentals

A naïve way to process the query is to expand its window so that it includes all the area that the query will cover up to a timestamp t in the future, and then process this extended window (using a regular R-tree window query) to find all candidate objects that may change the result up to time t. In the example of Fig. 1a, the extended window (bold rectangle) corresponds to the area that the query will cover in the next $t = 4$ time units. For all candidate objects (b, d, e), the interval during which they belong to the result is computed: for b this interval is [0, 1), for d it is [2, 4), and for e [3, 4). Given this information, we can determine the conventional and the TP components of the query. This method, however, has some serious shortcomings. The estimation t of how long in the future to extend the query window is ad-hoc. Under-estimation means that we will not be able to compute the time-parameterized component, while over-estimation will incur significant computational overhead. Additionally, the method is not applicable to other types of queries such as NN.

Observe that the result of a spatial query changes in the future because some objects "influence" its correctness. For instance, if an object (e. g., b) satisfies the query at the current time, it may influence the result when it no longer satisfies the query predicate in the future (at time 1). On the other hand, an object not currently in the result (e. g., d) may influence the query when it becomes a part of the result (at time 2). Figure 1a shows the influence time of all objects. Some objects, such as a and c, may never change the result, in which case the influence time is ∞.

The concept of "influence time" also applies to other types of queries. Consider Fig. 1b again where, as mentioned earlier, point d is the current nearest neighbor of q. In this case, the influence time of an object should be interpreted as the time that it starts to get closer to the query than d. For example, the influence time of point g is 3, because right after this time, g will come closer to q than d. Notice that a finite influence time does not necessarily mean that the object will change the result; g will influence the query at time 3 only if the result does not change before due to another object (actually at time 3, the nearest neighbor is object f). The influence time of points a, b, c is ∞ since they can never be closer to q than its current nearest neighbor d (observe that the influence time of d is also set to ∞). Let us denote the influence time of an object o with respect to a query q as $T_{INF}(o, q)$. The expiry time of the current result is the minimum influence time of all objects. Therefore, the time-parameterized component of a TP query can be reduced to a nearest neighbor problem by treating $T_{INF}(o, q)$ as the distance metric: the goal is to find the object(s) (i. e., the set **C**) with the minimum T_{INF} $(= \mathbf{T})$. These are the candidates that may generate the change of the result at the expiry time (by adding to or deleting from the previous answer set).

The state-of-the-art algorithms [4,5] for solving a NN query deploy a data-partitioning access method such as an R-tree [1] or a TPR-tree [7,11]. Application of these algorithms requires the formulation of *mindist*, the minimum "distance" from an intermediate entry E to the query q. In the context of TP retrieval, the *mindist*, denoted as $T_{INF}(E,$



a

	A_1	A_2	A_3	A_4	A_5
B_1	3	∞	∞	∞	∞
B_2	∞	∞	1	∞	∞
B_3	∞	∞	∞	4	2

b

Queries in Spatio-temporal Databases, Time Parameterized, Figure 2 Influence time of object pairs in a TP join. **a** Two sets of objects, **b** Influence time of all pairs

q), corresponds to the minimum influence time $T_{INF}(o, q)$ of any object o that may lie in the subtree of E. The detailed derivation of $T_{INF}(o, q)$ and $T_{INF}(E, q)$ depends on the concrete query types, as elaborated in [10].

A join query returns all pairs of objects from two datasets that satisfy some spatial condition. Consider Fig. 2a, which demonstrates two sets of objects: the first set involves static rectangles $A_1, A_2, \ldots, A_5$, and the second includes moving objects B_1, B_2, and B_3. Assuming "intersection" to be the join predicate, the current result involves three qualifying pairs: $\{(A_1, B_1), (A_3, B_2), (A_4, B_3)\}$. The join result changes in the future when (i) a pair of objects in the current result ceases to intersect, or (ii) a pair not in the result starts to satisfy the condition. Based on this observation, the notion of influence time is naturally applicable to any object pair in the Cartesian product of the two datasets. For instance, objects A_3 and B_2, which do not intersect at the current time, will start intersecting at time 1; hence, the influence time $T_{INF}(A_3, B_2)$ equals 1. Figure 2b lists the T_{INF} for all pairs of objects. The influence time is ∞ if a pair will never change the join result (e. g., (A_2, B_2)). The expiry time of the current result is the minimum influence time $T_{INF}(A_3, B_2) = 1$. As in the other types of TP queries, by adding or deleting the pair of objects (A_3, B_2) that causes the change, the join result can be updated incrementally at time 1.

A TP join can be regarded as a closest pair query [3] by treating $T_{INF}(o_1, o_2)$ as the distance metric between objects o_1 and o_2, and thus, processed using the previous CP solutions (see [10] for details).

Key Applications

A TP search is the building block of most continuous spatiotemporal queries. As an example, consider that a user driving down a highway wants to find all hotels within a 5 km range from her/his current position. In addition to a set of hotels (say A, B, C) currently within the 5 km range, the result contains the time (e. g., 1 minute) that this answer set is valid (given the direction and the speed of the user's movement) as well as the new answer set after the change (e. g., at 1 minute hotel D will start to be within 5 km). In the previous example, it is assumed that the query window is dynamic and the database objects are static. In other cases, the opposite may be true, e. g., find all cars that are within a 5 km range from hotel A. It is also possible that both the query and the objects are dynamic, if for instance, the query and the database objects are points denoting moving airplanes. The same concept can be applied to other common query types, e. g., nearest neighbors (keep a driver updated of her/his nearest gas station), and spatial joins (find all major residential areas currently covered by typhoons together with the earliest time that the situation is expected to change).

Future Directions

The concept of time parameterization is inherent in any dynamic scenario where the value of an entity is subject to continuous modification. For example, in sensor monitoring, one could formulate a TP query such as "report the sites in Washington D.C. where the temperature is above 90 F (for detection of fire hazards) as well as the earliest time when the result is expected to change". Efficient processing of such a query requires (i) an accurate model to predict the subsequent readings of a sensor (analogous to the linear model for capturing a moving object's future locations), and (ii) a specialized access method for minimizing query costs. Solutions to these problems remain largely open in the literature of sensor networks as well as other areas (streams, stochastic databases, uncertain data retrieval, etc.) where TP reasoning is applicable.

Q

Recommended Reading

1. Beckmann, N., Kriegel, H.P., Schneider, R., Seeger, B.: The R*-tree: An Efficient and Robust Access Method for Points and Rectangles. ACM SIGMOD, Atlantic City, NY, USA 23–25 May 1990
2. Chen, J., DeWitt, D.J., Tian, F., Wang, Y.: NiagaraCQ: A Scalable Continuous Query System for Internet Databases. ACM SIGMOD, Dallas, Texas, USA, 16–18 May 2000
3. Corral, A., Manolopoulos, Y., Theodoridis, Y., Vassilakopoulos, M.: Closest Pair Queries in Spatial Databases. ACM SIGMOD, Dallas, Texas, USA, 16–18 May 2000
4. Mokbel, M., Xiong, X., Aref, W.: SINA: Scalable Incremental Processing of Continuous Queries in Spatio-temporal Databases. ACM SIGMOD, Paris, France 13–18, June 2004
5. Samet, H., Hjaltason, G.: Distance Browsing in Spatial Databases. ACM TODS, 1999
6. Roussopoulos, N., Kelly, S., Vincent, F.: Nearest Neighbor Queries. ACM SIGMOD, San Jose, California, USA, 22–25 May 1995
7. Saltenis, S., Jensen, C., Leutenegger, S., Lopez, M.: Indexing the Positions of Continuously Moving Objects. ACM SIGMOD, Dallas, Texas, USA, 16–18 May 2000
8. Song, Z., Roussopoulos, N.: K-Nearest Neighbor Search for Moving Query Point. SSTD, Redondo Beach, CA, USA, 12–15 July 2001
9. Sistla, P., Wolfson, O., Chamberlain, S., Dao, S.: Modeling and Querying Moving Objects. IEEE ICDE, Birmingham, UK, 7–11 Apr 1997
10. Tao, Y., Papadias, D.: Time-Parameterized Queries in Spatio-temporal Databases. ACM SIGMOD, Wisconsin, 3–6 June 2002
11. Tao, Y., Papadias, D., Sun, J.: The TPR*-Tree: An Optimized Spatio-temporal Access Method for Predictive Queries. VLDB, Berlin, Germany 9–12 Sept 2003
12. Zheng, B., Lee, D.: Semantic Caching in Location-Dependent Query Processing. SSTD, Redondo Beach, CA, USA, 12–15 July 2001

Query-by-Content

► Indexing and Mining Time Series Data

Query, Datalog

► Constraint Databases, Spatial

Query Languages for Moving Objects

► Moving Object Languages

Query, Nearest Neighbor

► iDistance Techniques

RADAR

► Evolution of Earth Observation

Radio Frequency Identification (RFID)

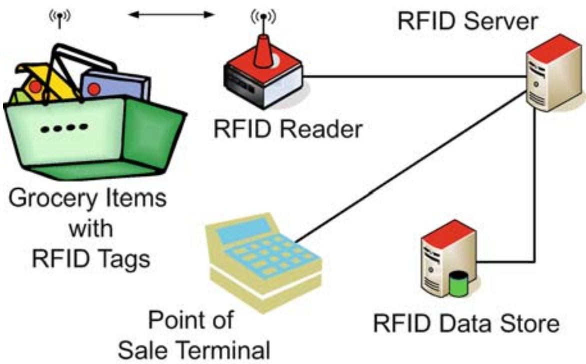
MICHAEL GOSHEY
Department of Computer Science and Engineering,
University of Minnesota, Minneapolis, MN, USA

Synonyms

RFID; RF identification; Electronic identification; Radio tagging; Electromagnetic tagging

Definition

Radio Frequency Identification (RFID) exists as a distinct subset of the larger family of automated identification technologies that includes things like bar codes, visual scanning devices and biometric readers. RFID is a means of automated identification that features electronic tags used both to store data and to act as transponders for sending the stored data as the payload in electromagnetic waves (radio waves) sent to detached listening devices (RFID readers) [1,2]. The tags can be affixed to animate or inanimate objects by a variety of methods and the readers that receive emissions from the tags translate the wave-embedded data into meaningful information (Fig. 1). They are a significant improvement over bar codes, for example, in that they do not require any human intervention. Currently deployed RFID systems provide real-time identity tracking and monitoring and make possible a wide variety of access control and inventory management solutions [3,4,5]. Beyond this primary value of real-time identification monitoring, the back-end of a typical RFID solution offers a distinct secondary value through storing the rich set of identity data captured by the system, enabling sophisticated data mining techniques to be applied to a number of interesting scenarios [6]. Additionally, while RFID systems primarily deal with automating identity management, they are frequently deployed in combination with Geographic Information Systems (GIS) to yield even more powerful solutions that track both identity and spatial location [7,8].



Radio Frequency Identification (RFID), Figure 1 Simple system diagram of grocery store point of sale RFID

Historical Background

While the practical application of Radio Frequency Identification has grown tremendously in recent years, the technology has a long and interesting history.

1920's–1940's

The seminal paper in the field is considered by many to be *Communication by Means of Reflected Power* by Harry Stockman in 1948 [9]. This paper proposed the key RFID notion: that power generated at the base (often fixed) end of a point-to-point electronic communication pair is reflected and reused in order to power the return transmission from the remote (often mobile) end of that pair. Stockman's paper echoed work that was already being done in *radar* to track the location of an object through the use of reflected radio waves which began in the early 1920's [1]. The 1930's and 1940's saw steady development in both the radio and radar fields, paving the way for Stockman's work, which was also reflected in the progress of another prominent technology of the era- Identification Friend or Foe (IFF) systems [10]. IFF systems employed transponders in attempts to avoid 'friendly fire' situations and were early predecessors of today's air traffic control systems. They were utilized during World War II to gain military advantage by identifying the allegiance of approaching aircraft before visual confirmation was possible [11,1].

1950's–Present

The 1950's saw the first work combining microwaves with RFID [12]. In the 1960's and 1970's the field gradually advanced on multiple fronts: academic research, patent activity and commercialization [1]. There were a number of patents related to remote measurement, communication and activation using radio frequency power, perhaps the most well-known of which was Charles Walton's 1973 patent for keyless door entry using passive RFID [13]. This was an especially important year for RFID as it also saw the first patent dealing with active RFID systems where tag memory could be rewritten and updated [14]. A couple of years later a landmark paper was authored by a group at Los Alamos [15], where work was being conducted on animal tracking and automated vehicle control systems. Large companies such as Phillips, Westinghouse, Raytheon and General Electric were ramping up work on shipping, transportation and factory automation systems based on RFID-related technologies. Automated toll-collection systems were in place both in the United States and in Europe by the mid-1980's and ultra-high frequency (UHF) RFID was developed in the early 1990's by IBM [14]. A team from MIT sponsored by Gillette and Proctor & Gamble later carried UHF-based RFID further, applying it to low-cost supply chain management applications that featured dumb tags containing only identification numbers rather than rewritable memory. Finally, in recent years it has been mega-merchant Wal-Mart (along with fellow retailers Target and Tesco and the U.S. Department of Defense) that has had perhaps the most influential hand in pushing RFID to the forefront of the global technology landscape [5]. Recent mandates by these large organizations to convert all of their vendors and suppliers to RFID-based supply chain management systems have impacted entire industries while raising significant concerns over the potential intrusiveness and other privacy drawbacks now associated with RFID [16,17].

Scientific Fundamentals

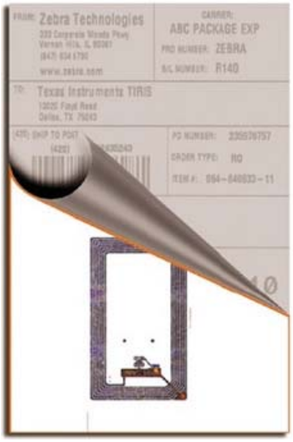
A good first step toward understanding a Radio Frequency Identification system is to inspect its primary components. RFID systems are supported through a network of application software, middleware and computing resources. While these play a role in the overall delivery of RFID and are perhaps specialized for this purpose they are not unique to the RFID domain. However at the center of these general actors are several RFID-specific components: tags, readers and label printers [33,34].

Tags

RFID systems exist for the purpose of capturing information about entities of interest (i. e. things one wishes to identify, track and monitor). In an RFID environment this information is stored on electronic tags. An RFID tag is simply a silicon microchip bundled with an antenna and attached to an entity of interest (Fig. 2). Not surprisingly the chips store identifying data such as unique serial numbers and other information required by the ID system. The tags act as transponders, receiving and responding to radio signals to and from RFID readers. RFID has some distinct advantages over bar code technology, including the ability of tags to function without line of sight and the ability of readers to process hundreds of chip reads virtually simultaneously. There are three main types of RFID tags: Passive, Active and Semi-Passive.

Radio Frequency Identification (RFID), Figure 2 RFID chip surrounded by its long antenna coil. (Source: Defense Logistics Agency (DLA))

Passive Tags Passive RFID tags do not have their own power supply. They harvest necessary power from the electromagnetic fields emanating from the incoming radio wave transmissions they receive from RFID readers. As transponders they 'awake' on demand by the reader's signal and induce a small current in the on board antenna to power-up the internal (CMOS) microchip and send back data to the reader. The term 'backskatter' is used to describe the process by which the tags reflect the signals they receive from readers as their message-laden reply. The reflected signals are modulated variations of the signals that are received and therefore produce meaningful data- the difference between the original and the (modulated) variant. Today's chips hold up to about two kilobytes of data, enabling the storage of rich information about the tagged entity. Because passive tags do not require on-board batteries they have a significantly longer life expectancy



Radio Frequency Identification (RFID), Figure 3 Passive RFID label. (Source: DLA)

than active tags, can be made much smaller and are therefore cheaper to produce. Unit costs vary with capabilities and purchase volume but the lowest cost passive tags currently sell for about twenty U.S. cents per unit in high volume purchases. New non-silicon (polymer) roll-printable tags are currently being prototyped and these should further reduce the manufacturing cost. Typical sizes range from postage stamps to postcards though the smallest passive tags are thinner than a standard piece of paper and approximately three millimeters across and therefore can be inserted under the skin of animals and people or fit nicely on a sticker or paper label (Fig. 3). The lack of internal power creates a limitation on broadcast distance and thus passive tags are generally used for short distance transmissions (under 10 feet). Passive tags are typically write-once-read-many (WORM) devices, meaning that their memory cannot be overwritten.

Active Tags Unlike their passive counterparts that employ backscatter to send message replies to readers, active tags have an internal transmitter that allows them to send waves under their own power and conduct what amounts to 'sessions' with readers. Most have on-board batteries, though some use solar and other power supply options to harvest energy to power the transmitter. The resulting larger form factor and higher cost is offset by the distinct benefit of being able to transmit stronger signals, useful in situations where required read distances are much further as well as in environments that are difficult for passive tags to function well (such as those high in metals or water). Active tags often have readable ranges of hundreds of meters and vary in price from a few dollars up to twenty dollars or more per unit. Life expectancy of active RFID chips tends to be based on battery life and is typically up to 10 years. In addition to environments that challenge passive tags, active RFID tags are often used to track valuable assets. They usually have more data storage capability than passive tags and may also provide read-write capabilities so that internally stored data can be updated and changed. Active tags are frequently combined with environmental sensors to produce sophisticated monitoring devices that can, for example, track a product in its shipping route while monitoring the temperature and other conditions in which it is being transported.

Semi-Passive Tags Semi-passive tags are a hybrid of active and passive tags. They have on-board battery power like active tags but they use backscattering to reflect radio waves back toward the readers that produce them, rather than transmitting their own message broadcasts. The battery is used to power the microchip and (optionally) environmental sensors. While their versatility exceeds that of passive tags they are cheaper than active tags and their low battery consumption means they live longer.

Readers

In an RFID system the readers are the components that directly communicate with and fetch data from tags. Readers may be stationary or mobile (Fig. 4) and have one or more antennas which broadcast signals to nearby tags, querying information stored in the embedded chips as well as (occasionally) from on-board sensors. Agile readers are highly functional and can read tags at different frequencies and use multiple communication protocols in communicating with various tags. Some readers (especially those

Radio Frequency Identification (RFID), Figure 4 Mobile RFID reader with tags in foreground. (Source: PM J-AIT) in the ultra-high frequency range) can have a 'null spot', a blind-spot of sorts for an RFID reader. There are two primary types of RFID readers: **read-only** and **read-write**. As implied by the name, the former class is limited to reading and extracting data from tags. Some in this category are referred to as 'dumb' due to limited computing capabilities. Dumb readers convert the radio waves they receive from RFID tags into binary values and simply forward the converted values to a server for processing of actual business logic and filters. Unlike read-only devices, the latter class of readers can read from RFID tags as well as write to them. The roles of read-write readers include initializing chips on new WORM tags and writing to updateable chips during RFID transactions.

Air Interface Protocol In any field the work of establishing and refining protocols and standards is long term and ongoing. An area of obvious focus in RFID is the communication between readers and tags. Whether articulating the differences between *tag talks first* (new tags crossing a reader's threshold 'announce' themselves to the reader) and *reader talks first* (readers initiate contact with tags by sending out inquiries to all listening tags) protocols, following mutual encoding rules, establishing system commands for reading and writing data or knowing the rules for how tags should modulate responses back to readers, the standards and protocols are critical to the success of RFID. Among the many standards governing the RFID industry the air interface protocol provides specific directives concerning the communication between readers and tags. As of July 2006 the new 'Gen 2' air-interface protocol was incorporated into the ISO 18000–6 standard [19].

Singulation and Anti-Collision One of the specific technical challenges of RFID readers is identifying unique tags amongst many tags within a given read field and being able to gracefully handle the resulting collisions from simultaneous radio responses by multiple tags (singulation). This is a common problem in environments such as warehouses and grocery stores, where RFID readers routinely have hundreds of tags within close proximity. One commonly used singulation protocol is known as 'tree walking.' The RFID reader logically traverses a binary tree in a recursive, depth-first algorithm. Each node has only two children: 0 for the left subtree and 1 for the right subtree (the root node is considered empty). The reader broadcasts each successive bit sequence (i. e. tree node label) to all of the tags within its range, instructing only those tags with serial numbers matching the broadcast sequence to respond (by providing their next serial number bit), while the rest of the tags remain silent. When collisions result between responses from both the left and right subtrees of the broadcast node, the reader detects the collision and interprets it as a requirement to descend into both subtrees (recursively and sequentially). The algorithm ensures that the reader descends into every required subtree (i. e. those containing currently present leaf nodes). As the reader recurses through the tree, it captures the leaf nodes bit strings, resulting in a list of all (binary) serial numbers of the individual tags in its range [18]. Though it is widely used, the tree-walking protocol exposes RFID systems to substantial threats of privacy and security. The identify of tags within range of the reader can be deduced by eavesdropping on only one side of the electronic conversation (the reader side) and the longer broadcast range of readers enables such eavesdropping to be conducted at distances well beyond the range of the tags themselves.

Label Printers

The RFID industry distinguishes between tags and labels: transponders mounted to a substrate for the purpose of attaching to a carton or other surface are considered RFID tags. Transponders with the smallest (thinnest) form factor that are bound with adhesive between pieces of paper in order to be used in labels are called RFID labels. RFID label printers are highly specialized printing machines that output 'smart' labels which combine bar code printing technology with RFID. The printers simultaneously print bar codes on the front of the labels while writing (related) data to the on-board chips of RFID transponders embedded between the sheets of label paper (Fig. 5).

Radio Frequency Identification (RFID), Figure 5 Smart label printer combining bar codes and RFID tags. (Source: DLA)

Key Applications

RFID is broadly utilized across a wide array of identification systems, addressing such problems as inventory control, tracking and supply chain management, passport, currency and border control, transportation access and automated fare/toll collection, hands-free ski lift access, bovine control, wildlife tracking and pet identification, building access control, performance monitoring and sensor reading, prisoner tracking, hospital bed management and crime (theft, tampering, counterfeiting) deterrence. The technology has become ubiquitous in modern society and is embedded in the products sold by mass retailers, the currency and credit cards used for payment, vending machines, library systems, warehouse pallets, airline baggage handling machines, automobile tires, keys, sports match timing devices, animals and people.

Radio Frequency Identification (RFID), Figure 6 Truck transponder sends data to roadside RFID reader via overhead antennae. (Source: FHWA)

Inventory Control

In 2003 mega-retailer Wal-Mart single-handedly initiated a major industrial trend when it announced it would be requiring top suppliers to implement RFID technology for inventory management. Others were quick to follow the world's largest company down the RFID path, including the Department of Defense, Target Stores, Best Buy, Kroger, Circuit City and others. Prior to this, one of the biggest obstacles to adoption was the relatively high cost of producing tags but by throwing its (unparalleled) business volume behind the technology Wal-Mart played a significant role in driving down cost and ensuring RFID's long term viability. In inventory management systems RFID tags are affixed to pallets, boxes and even individual products to facilitate inventory tracking with little required human intervention. For example entire cartons of products can be added to the inventory pool without requiring they be scanned individually or even removed from the box. In the long term, RFID may also be used to track the movement of products through stores. Though Wal-Mart has lagged behind its original goal of having its 100 top suppliers converted by January 2005 it continues to push ahead and recently announced it would double the number of its stores that have implemented RFID and that the number would reach 1000 locations by the beginning of 2007 [35].

Supply Chain Management

Wal-Mart's introduction of RFID into the retail space was echoed by a similar move at the U.S. Department of Defense (DOD) to transition its supply chain management structure to an RFID-based system. Over the past several years the DOD has implemented RFID on a broad, international scale across its operations. One of the interesting RFID issues in supply chain management is tag ownership. Unlike in-house applications where a single entity invests in and maintains a set of tags for RFID, a supply chain by definition contains many actors and goods pass through many boundaries of ownership and control as they traverse the chain from source to destination (Fig. 6). In such an environment, tags simply become part of the shipping material and are rarely recycled or reused by the originating party. These are precisely some of the requirements driving tagging technology toward extremely low cost tags that can become as ubiquitous and disposable as bar codes are today. Tag disposal and recycling are related concerns that take on new meaning in such cooperative systems. Finally, just as the lines of tag ownership blur in supply chain applications, so too does the ownership of technology standards. Privately controlled proprietary standards do not meet the requirements of systems punctuated by repeated hand-offs from one party to another. Applications such as supply chain management will compel the state of RFID technology to continue to move toward open industry standards.

Passport, Currency and Border Control

Another key RFID application is in the area of protecting national assets such as passports, currency and physical borders. RFID passport control systems already exist in many countries. The U.S. has had an off-again on-again relationship with passport-embedded RFID but as of October 2006 is now issuing all new passports with ISO 14443 RFID tags. Security is an obvious and critical requirement for such systems yet this is an area where RFID has struggled. In the Netherlands in early 2006 the Dutch security company Riscure demonstrated a successful compromise of a Dutch RFID-passport based on the same ISO standard [36]. Even human-embedded RFID tags have been unable to escape security controversy. In July of 2006 hackers at a conference in New York City demonstrated the cloning of an 'uncloneable' implanted RFID chip and successfully fooled the RFID reader concerning the identity of the bearer [22]. Another area of broad concern has been the tracking of national currencies for the purposes of reducing fraud and criminal enterprise. While the complete materialization of a plan may be years away, there has been a great deal published in both academia and industry on the specter of embedded RFID tags in European currency [21]. Finally, while contact-less toll collection systems have been in place for many years, they are now being applied to certain border crossings between the US and Canada. Completely automated border security may not be realistic, but in such cases RFID is facilitating speedy identification of those wishing to pass national boundaries.

Future Directions

As seen in the previous section, radio frequency identification is a technology with seemingly limitless potential for future growth. When combined with GIS, that landscape of future possibilities appears to grow without bound. Systems are being pioneered and developed today that bring the two technologies together in interesting ways such as in wildlife management, military field training and exercises, monitoring of children through embedded RFID/GIS devices in school bags, automated manhole monitoring and intelligence, remote forestry management, election support and targeted voting campaigns, 'pervasive retail' and 'smart' environments that are aware of and respond to entities passing through them. However, as RFID and GIS converge with other technologies such as bar coding, digital cameras and biometrics, concerns about security and privacy will continue to grow. Issues such as data privacy and identity theft, RFID tag blocking, data leaking and RFID viruses will grow in prominence and public awareness. RFID's impact on the environment in such areas as device disposal and increased radio wave transmissions will likely become a major area of emphasis, as will increased efforts toward further standardization of worldwide RFID systems. Finally, one would expect that as systems begin to mature and require less break-fix attention the systematic mining of considerable amounts of newly captured data will likely become a major area of increased research and emphasis.

Cross References

- ► Emergency Evacuation Plan Maintenance
- ► Emergency Evacuations, Transportation Networks
- ► Homeland Security and Spatial Data Mining
- ► Information Services, Geography
- ► Privacy and Security Challenges in GIS

Recommended Reading

1. Landt, J.: Shrouds of Time: The History of RFID. http://www.aimglobal.org/technologies/rfid/resources/shrouds_of_time.pdf, Association for Automatic Identification and Mobility: Retreived 10/10/2006 (2001)
2. Wikipedia: Radio Frequency Identification. http://en.wikipedia.org/w/index.php?title=Radio_Frequency_Identification&oldid=85251329, Wikipedia, The Free Encyclopedia: Retreived 10/10/2006 (2006)
3. Vogt, H.: Pervasive Computing: First International Conference Proceedings, In: Efficient Object Identification with Passive RFID Tags, Lecture Notes in Computer Science, vol. 2414, pp. 98. Springer (2002)
4. The British Broadcasting Corporation: Stopping the book thieves. http://news.bbc.co.uk/2/hi/uk_news/1958424.stm, The BBC News: Retreived 10/10/2006 (2002)
5. Shim, R.: Wal-Mart to throw its weight behind RFID, http://news.com.com/2100-1022-1013767.html?tag=nl, CNET News: Retreived 10/10/2006 (2003)
6. Bauer, M., Fabian, B., Fischmann, M., Gürses, S.: Emerging Markets for RFID Traces, Arxiv preprint cs.CY/0606018 (2006)
7. Bluestein, G.:, Companies track gridlock via cell phones. http://news.yahoo.com/s/ap/20061105/ap_on_hi_te/tracking_traffic;_ylt=AoQQdZOHq.vxPTOcxWTRCaDMWM0F;_ylu=X3oDMTA3ODdxdHBhBHNlYwM5NjQ-, Associated Press: Retreived 11/05/2006 (2006)
8. Saxena, M.: Does GIS and RFID Amalgamation Work? http://www.geoplace.com/uploads/FeatureArticle/0505tt.asp, Geoplace: Retreived 10/10/2006 (2005)
9. Stockman, H.: Communication by means of reflected power, Proc. Inst. Radio Engin. **36**, 1196–1204 (1948)
10. The United States Navy: Identification Friend or Foe. http://www.nrl.navy.mil/content.php?P=FRIENDFOE, United States Naval Research Library: Retreived 11/05/2006 (2006)
11. Bowden, L.: The story of IFF (identification friend or foe), IEEE Proceedings, Part A: Physical Science, Measurement and Instrumentation, Management and Education **132**(6), 435–437 (1985)
12. Vernon Jr, F.: Application of the microwave homodyne, Antennas and Propogation, Transactions of the IRE Professional Group on **4**(1), 110–116 (1952)
13. Walton, C.A.: US Patent 3,752,960, 27 December 1971
14. The RFID Journal: The History of RFID Technology. http://www.rfidjournal.com/article/articleview/1338/1/129/, The RFID Journal: Retreived 10/10/2006 (2006)
15. Koelle, A.R., Depp, S.W., Freyman, R.W.: Short-range radiotelemetry for electronic identification, using modulated RF backscatter, Proc. IEEE **63**(8), 1260–1261 (1975)
16. Sanjay E. Weis, S., Weis, S.A., Engels, D.W.: RFID Systems and Security and Privacy Implications, In: Workshop on Cryptographic Hardware and Embedded Systems, Lecture Notes in Computer Science, vol. 2523, pp. 454–470 (2002) http://citeseer.ist.psu.edu/sarma02rfid.html
17. Molnar, D., Wagner, D.: Privacy and security in library RFID: issues, practices, and architectures, In: CCS '04: Proceedings of the 11th ACM conference on Computer and communications security, pp.210–219. ACM Press, New York (2004)
18. Juels, A., Rivest, R.L., Szydlo, M.: The blocker tag: selective blocking of RFID tags for consumer privacy, In: CCS '03: Proceedings of the 10th ACM conference on Computer and communications security, pp. 103–111. ACM Press, New York (2003)
19. O'Connor, M.C.: Gen 2 EPC Protocol Approved as ISO 18000–6C. http://www.rfidjournal.com/article/articleview/2481/1/1/, The RFID Journal: Retreived 11/05/2006 (2006)
19. Avoine, G., Oechslin, P.: Financial Cryptography and Data Security, In: RFID traceability: A multilayer problem, Lecture Notes in Computer Science, vol. 3570, pp. 125–140. Springer (2005)

21. Juels, A., Pappu, R.: Financial Cryptography, In: Squealing Euros: Privacy Protection in RFID-Enabled Banknotes, Lecture Notes in Computer Science, vol. 2742, pp. 103–121. Springer (2003)
22. Fulton, N.: High-tech cloning. http://blogs.reuters.com/2006/07/22/high-tech-cloning/, Reuters: Retreived 10/10/2006 (2006)
23. Weis, S.A., Sarma, S.E., Rivest, R.L., Engels, D.W.: Security in Pervasive Computing, In: Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems, Lecture Notes in Computer Science, vol. 2802, pp. 201–212. Springer (2004)
24. Indulska, J., McFadden, T., Kind, M., Henricksen, K.: Distributed Applications and Interoperable Systems, In: Scalable Location Management for Context-Aware Systems, Lecture Notes in Computer Science, vol. 2893, pp. 224–235. Springer (2003)
25. Roussos, G., Kourouthanasis, P., Spinellis, D., Gryazin, E., Pryzbliski, M., Kalpogiannis, G., Giaglis, G.: Systems architecture for pervasive retail, In: SAC '03: Proceedings of the 2003 ACM symposium on Applied computing (ACM Press, New York 2003)
26. Newsweek: An Internet of Things. http://www.msnbc.msn.com/id/3068871/, Newsweek: Retreived 10/10/2006 (2006)
27. Davies, N., Gellersen, H.W.: Beyond prototypes: challenges in deploying ubiquitous systems, Pervasive Computing, IEEE **1**(1), 26–35 (2002)
28. Tenmoku, R., Kanbara, M., Yokoya, N.: A wearable augmented reality system using positioning infrastructures and a pedometer, Wearable Computers, 2003. Proceedings. Seventh IEEE International Symposium on 110–117 (2003)
29. Want, R., Fishkin, K.P., Gujar, A., Harrison, B.L.: Bridging physical and virtual worlds with electronic tags, In: CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, New York (1999)
30. Coroama, V., Bohn, J., Mattern, F.: Living in a Smart Environment – Implications for the Coming Ubiquitous Information Society, In: Proceedings of the International Conference on Systems, Man and Cybernetics 2004 (IEEE SMC 2004) 5633–5638, http://citeseer.ist.psu.edu/coroama04living.html
31. Juels, A., Weis, S.A.: Defining Strong Privacy for RFID. http://eprint.iacr.org/2006/137.pdf, Cryptology ePrint Archive: Retreived 10/10/2006 (2006)
32. Roberti, M.: DOD Releases Final RFID Policy. http://www.rfidjournal.com/article/articleview/1080/1/1/, The RFID Journal: Retreived 10/11/2006 (2004)
33. The RFID Journal: The History of RFID Technology. http://www.rfidjournal.com/article/articleview/1337/1/129/, The RFID Journal: Retreived 10/12/2006 (2006)
34. The RFID Journal: RFID System Components and Costs. http://www.rfidjournal.com/article/articleview/1336/1/129/, The RFID Journal: Retreived 10/12/2006 (2006)
35. The RFID Gazette: Wal-Mart Doubling RFID-Enabled Stores. http://www.rfidgazette.org/walmart/, The RFID Journal: Retreived 11/05/2006 (2006)
36. Riscure: Privacy issues with new digital passport. http://www.riscure.com/2_news/passport.html, Riscure BV: Retreived 11/08/2006 (2006)

Radio Tagging

► Radio Frequency Identification (RFID)

Radiolocation

► Indoor Positioning with WirelessLocal Area Networks (WLAN)

Range Combining

► Indoor Localization

Range Query

► Nearest Neighbors Problem
► R*-tree

Range Query Algorithm

► Indexing of Moving Objects, B^x-Tree

Ranking, Multi-Dimensional

► Skyline Queries

Ranking Results

► Skyline Queries

Raster

► Spatial Data Transfer Standard (SDTS)

R

Raster Data

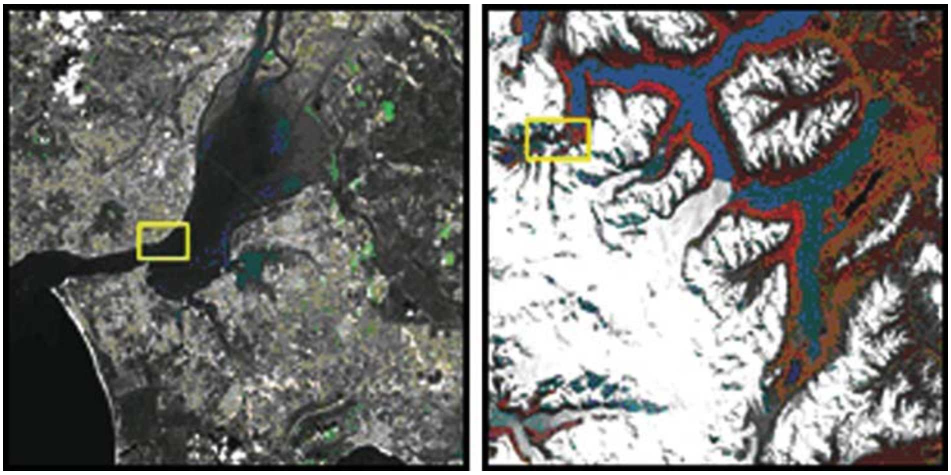
Hyeeun Lim
Department of Computer Science & Engineering, University of Minnesota, Minneapolis, MN, USA

Synonyms

Raster image; Digital image; Bitmap

Definition

Raster displays and databases build all geographic features from grid cells in a matrix. A raster display builds an image from pixels, or elements of coarse or fine resolution. A raster database maintains a similar "picture" of reality in which each cell records some sort of information averaged over the cell's area. The size of the cell may again be coarse or fine, ranging from centimeters to kilometers.



Raster Data, Figure 1 Imagery samples (5 m) from SPOT-5 [4]

Basically, a raster file is a giant table, where each pixel is assigned a specific value from 0 to 255. The meaning behind these values is specified by the user. They can represent elevations, temperatures, hydrography, etc. Satellite imagery uses raster data to record different wavelengths of light. Many satellites, like Landsat and SPOT (Fig. 1), transmit raster images of the earth's surface. Raster data is generally divided into two categories: thematic data and image data. The values in thematic raster data represent some measured quantity or classification of particular phenomena such as elevation, pollution concentration or population. For example, in a landcover map the value 5 may represent forest, and the value 7 may represent water. The values of cells in an image represent reflected or emitted light or energy such as that of a satellite image or a scanned photograph.

Historical Background

The earliest geographic information systems (GIS) architectures, implemented by Roger Tomlinson in the Canadian Land Inventory in the mid 1960s, emulated traditional map drafting. Entities were represented by points and lines that could be drawn with an automated drafting machine (aka pen plotter.) An outline history of GIS can be found at National Center for Geographic Information and Analysis History of GIS.

At The Harvard Lab for Computer Graphics and Spatial Analysis [at the Harvard Graduate School of Design (GSD)] in the late 1960s and early 1970s, Carl Steinitz and many others were experimenting with ways to use digital geographic data to emulate the cartographic overlay techniques portrayed by Ian McHarg in his book *Design with Nature*. McHarg and Steinitz collaborated in the first digitally augmented landscape planning studio at the GSD in 1967.

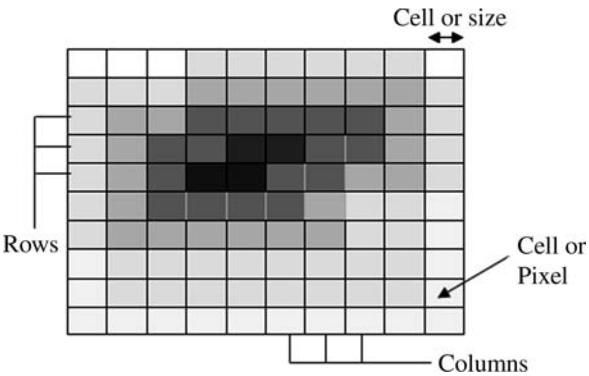
The students and researchers at Harvard were using a computer language called FORTRAN, IBM's highly customizable information tool to represent landscapes and things that may happen on them. In FORTRAN, one of the fundamental forms of representation is a matrix: an array of values. Naturally, someone began to experiment with this structure to represent the character of locations in 2-D space, and Raster GIS was born.

In general, raster data are more suited to environmental applications while vector data are more suited to human activity. Raster systems model complex spatial patterns with limited attributes, such as land-use patterns very well, while the vector data model is better for more clearly defined space with complex attributes, such as census data. There are exceptions to this: Martin (1996) uses derived raster surfaces to model 1981 and 1991 census data and compare change between the two, claiming that the raster model provides a more realistic model of the underlying population distribution. A good example of a raster system in a historical context is provided by Bartley and Campbell (1997). They examined the Inquisitions Post-Mortem of the fourteenth century and used these to create a raster GIS of medieval land-use that, they claim, is potentially the most detailed survey possible until the nineteenth century tithe surveys. A raster system was used because it provides a complete coverage of the land area, it provides a more realistic representation of land use than polygons, and because it handles source inaccuracies better than polygons.

Scientific Fundamentals

Raster Data Structure

Raster data types consist of rows and columns of cells, in each of which is stored a single value. Cells are arranged



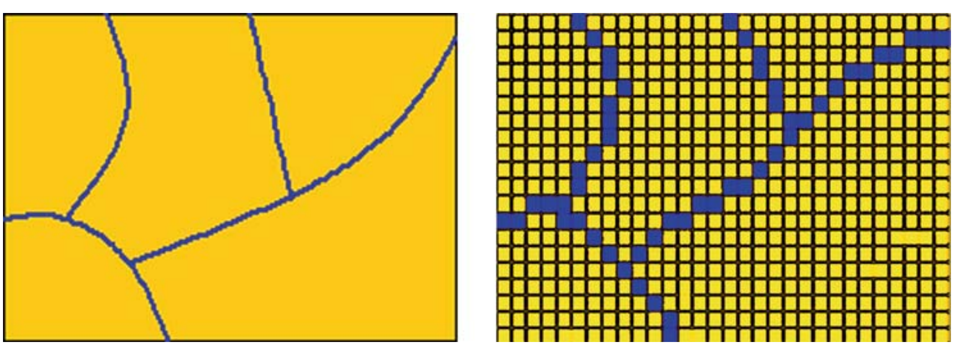
Raster Data, Figure 2 Raster data structure

in rows and columns where rows represent the *x*-axis of a Cartesian plane and the columns the *y*-axis. Each cell has a unique row and column address. Figure 2 shows the raster data structure. Each cell must be rectangular in shape, but not necessarily square. Each cell within this matrix contains location coordinates as well as an attribute value. Values can represent magnitude, distance, or relationship of the cell on a continuous surface. Values can also represent categorical data such as soil type or land-use class. Both integer and floating-point values are supported in spatial analyst. Integer values are best used to represent categorical data and floating-point values to represent continuous surfaces such as elevation, slope or flow accumulation. Rastering is a method to store, process and display spatial data. Most raster data are images, but in addition to color, the value recorded for each cell may be a discrete value, such as land use, a continuous value, such as rainfall, or a null value if no data is available.

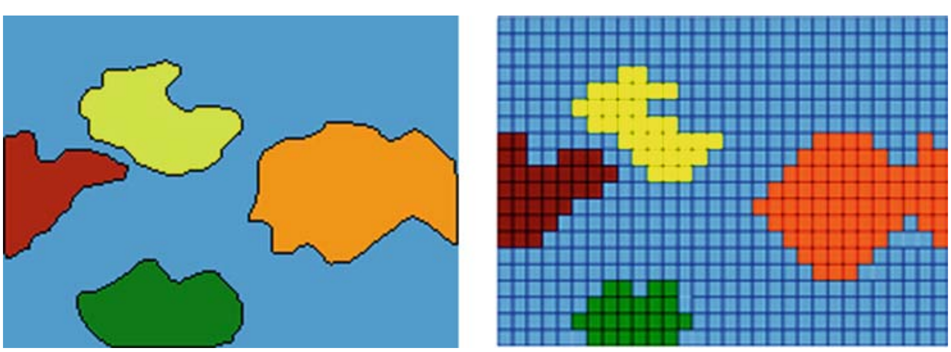
The spatial location of each cell is implicitly contained within the ordering of the matrix, unlike a vector structure which stores topology explicitly. Areas containing the same attribute value are recognized as such; however, raster structures cannot identify the boundaries of such areas as polygons. While a raster cell stores a single value, it can be extended by using raster bands to represent red, green and blue(RGB) colors, colormaps (a mapping between a thematic code and RGB value), or an extended attribute table with one row for each unique cell value. Thus, these two or more cells with the same value belong to the same zone. A zone can consist of cells that are connected, disconnected, or both. Zones are not analagous to polygons, which are only contiguous closed areas.

Representing Features in a Raster Dataset

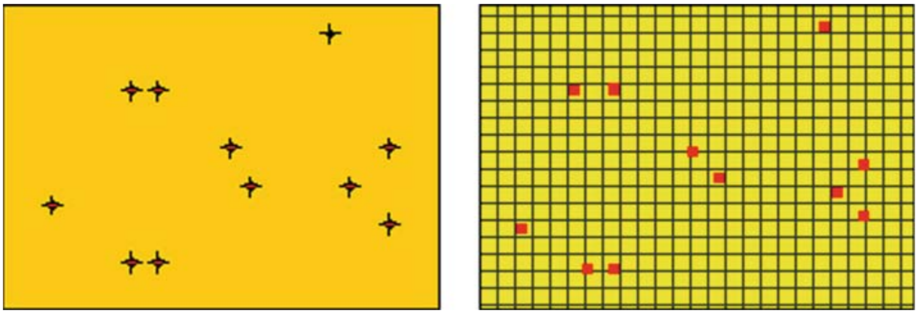
Linear Data Linear data is all of those features that, at a certain resolution, appear only as a polyline such as a road, a stream, or a power line. A line by definition does not have area. A polyline can be represented only by a series of connected cells (Fig. 3). As with a point, the



Raster Data, Figure 3 A set if line features represented in a grid



Raster Data, Figure 4 A set of polygon features represented in a grid



Raster Data, Figure 5 A set of point features represented in a grid

accuracy of the representation will vary according to the scale of the data and the resolution of the raster dataset.

Polygon Data Polygonal or aerial data is best represented by a series of connected cells that best portrays its shape (Fig. 4). Trying to represent the smooth boundaries of a polygon with a series of square cells does present some problems, the most infamous of which is called the "jaggies", an effect that resembles stair steps.

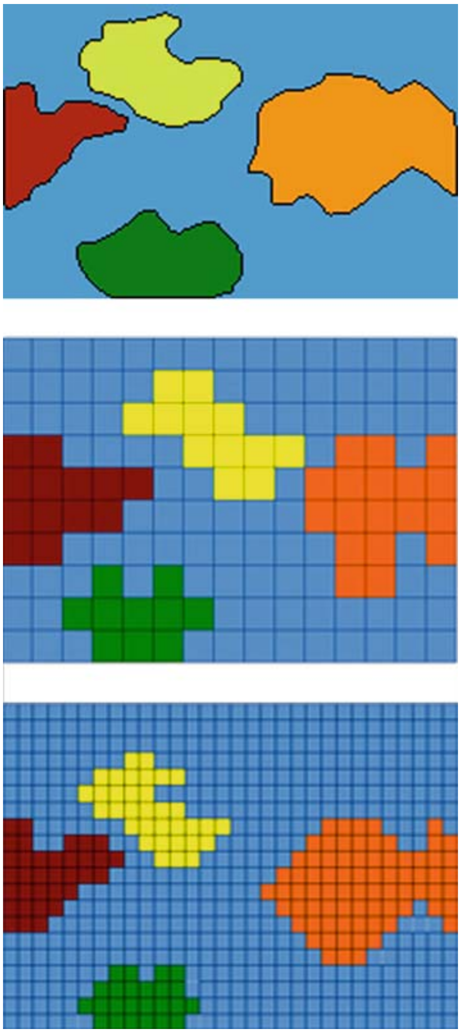
Point Data A point feature is any object at a given resolution that can be identified as being without area. Although a well, a telephone pole, or the location of an endangered plant are all features that can be rendered as points at some resolutions, at other resolutions they do, in fact, have area. Point features are represented by the smallest unit of a raster, a cell (Fig. 5). It is important to remember that a cell has area as a property. The smaller the cell size, the smaller the area and thus the closer the representation of the point features. Points with area have an accuracy of plus or minus half the cell size. This is the trade-off that must be made when working with a cell-based system. Having all data types – points, polylines, and polygons – in the same format and being able to use them interchangeably in the same language are more important to many users than a loss of accuracy. The accuracy of the above representation is dependent on the scale of the data and the size of the cell.

The Resolution of a Raster Dataset

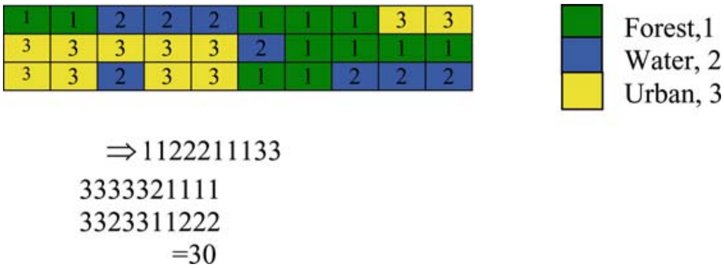
The size chosen for a raster cell of a study area depends on the data resolution required for the most detailed analysis (Fig. 6). The cell must be small enough to capture the required detail, but large enough so that computer storage and analysis can be performed efficiently. The more homogeneous an area is for critical variables such as topography and land use, the larger the cell size can be without affecting accuracy. Before specifying the cell size, the following factors should be considered:

- The resolution of the input data
- The size of the resultant database and disk capacity
- The desired response time
- The application and analysis that are to be performed

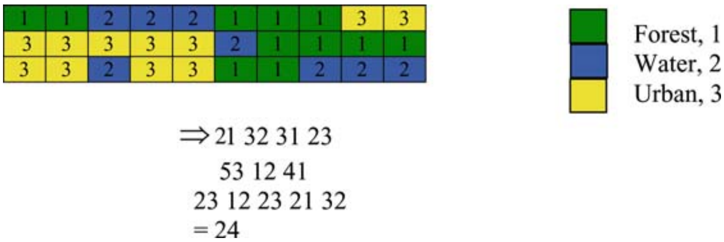
A cell size which is finer than the input resolution will not produce more accurate data than the input data. It is generally accepted that the resultant raster dataset should be the



Raster Data, Figure 6 An example of different resolution raster data maps



Raster Data, Figure 7 An example of exhaustive enumeration



Raster Data, Figure 8 An example of run-length encoding

same or coarser than the input data. Spatial analysis allows for raster datasets of different resolutions to be stored and analyzed together in the same database.

Raster Data Compression Technique

Exhaustive Enumeration Every pixel is given a single value; there is no compression when many like values are encountered (Fig. 7).

Run-Length Encoding This is a raster image-compression technique. If a raster contains groups of cells with identical values, run-length encoding can compress storage (Fig. 8). Instead of storing each cell, each component stores a value and a count of cells with that value. If there is only one cell, the storage doubles, but for three or more cells, there is a reduction. The longer and more frequent the consecutive values, the greater the compression that will be achieved. This technique is particularly useful for encoding monochrome images or binary images.

Raster Data Versus Vector Data

Vector Data The vector data type uses geometries such as points, lines or polygons to represent objects. Examples include property boundaries for a housing subdivision represented as polygons and well locations represented as points. Vector features can be made to respect spatial integrity through the application of topology rules such as polygons not being allowed to overlap. Vector data can also be used to represent continuously varying phenomena. Contour lines and triangulated irregular networks (TINs) are used to represent elevation or other continuously changing values. TINs record values at point locations, which are connected by lines to form an irregular mesh of triangles. The face of the triangles represents the terrain surface.

Comparison Between Vector Data and Raster Data See Table 1 and Fig. 9

Key Applications

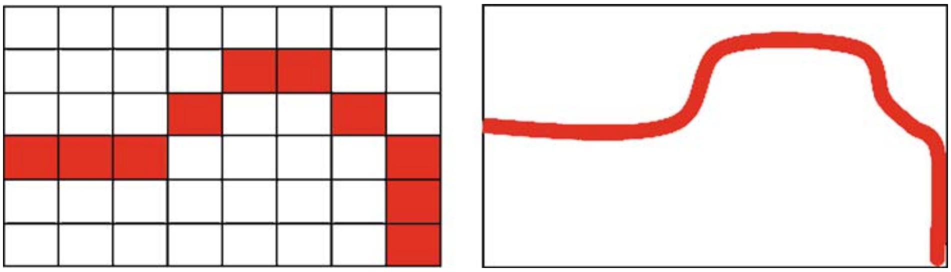
Business Mapping/Geomarketing

Business mapping improves the understanding and control of the market's realities and opportunities through the connections that it creates between data and the corresponding geographical location. Geomarketing, a derivative of business mapping, is the branch of marketing that helps to model and analyze a set of factors in order to find out correlations between a consumer's dwelling location and workplace, as well as his consumption patterns. Thematic raster data can be used for business mapping and geomarketing in order to optimize business actions by favoring geographical criteria through the use of statistical analysis and cartography (Fig. 10).

R

Health Geography

Raster data plays a major role in several areas of health geography. The cartographic representation of health data helps to describe and analyze a situation, and clarify it in complex cases. But the true purpose of health geography is to determine the reasons behind the phenomena. It is also important not to ignore the potential for new interactions with other fields of activity when geography is



Raster Data, Figure 9 A raster image compared with a vector image

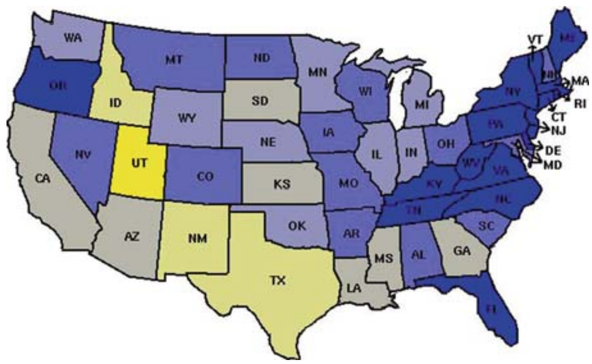
Raster Data, Table 1 Comparison between vector data and raster data

	Raster data	Vector data
Characteristics	Simple “grid” structure of rows and columns. Based on cells or picture pixels. Linear feature is a contiguous set of cells. Resolution based on size of grid (cell): the smaller the cell, the higher the resolution. Features are considered homogenous within a pixel. Storage increases with the square of the resolution	Based on objects (points, lines, areas). Constructed using arcs, nodes and vertices. Resolution can be independent of detail. Every point has a unique location
Advantages	A simple data structure. Overlay operations are straight forward. High spatial variability is efficiently represented Only raster can store image data (e. g., photos)	Compact data structure for homogenous areas. Efficient encoding of topology. Better suited for map output
Disadvantages	Data structure is not compact (though it can be modified). Topological relationships are harder to represent. Map output can appear “blocky”	More complex data structure. Overlay operations are more complex. High spatial variability is less efficiently stored. Cannot store (continuously varying) image data

used dynamically in the health care system since it adds demographic, economic, social, cultural and environmental dimensions. The application of raster thematic data to health geography for analysis and decision making is an area with great potential in today’s information and communication age.

Satellite Imagery

Satellite imagery is always in a raster format in which each pixel has its own signature. Reflectance at a certain wavelength is measured for each cell in an image. The cells may cover areas on the earth’s surface several hun-



Raster Data, Figure 10 An example of thematic maps

dreds of meters square, the area covered being a function of a particular satellite’s resolution. For example, Landsat, SPOT and digital raster graphics (DRG) are types of satellite data. Landsat, or thematic mapped data, was one of the first types of satellite data available to the public. It gives, at best, 10-m resolution although there is discussion that the new satellites to be launched soon will improve on that. Landsat, or thematic mapped data is very cost effective for large areas, particularly when we wish to acquire older scenes for an historic perspective. SPOT data is produced by a French company. It is a later form of the technology, and will give 3-m resolution. DRG data is generated from scanned quad maps. They can be very useful as backdrops to other data, for guiding digitizing, assisting in classification, or other operations where broad background coverage is needed.

Future Directions

Raster data is an abstraction of the real world where spatial data is expressed as a matrix of cells or pixels with spatial position implicit in the ordering of the pixels. Raster data can be applied to various areas such as business mapping, geomarketing, health geography or satellite imagery. Its usefulness for analysis and decision making will continue to grow in today’s information and communication age.

Cross References

► Information Services, Geography
► Vector Data

Recommended Reading

1. Wikipedia http://en.wikipedia.org/wiki/Geographic_information_system Accessed 23 March 2007
2. Introduction to GIS–Raster-Based GIS. http://www.sli.unimelb.edu.au/gisweb/GISModule/GIST_Raster.htm Accessed 21 February 2007
3. Understanding Raster Data And Cell-Based Modeling. http://www.cfr.washington.edu/research.urbaneco/GIS_Tutorial Accessed 21 February 2007
4. Geo VAR-Geospatial Data and Information System http://www.geovar.com/data/satellite/spot/spot.htm?gclid=CNHD6OS8ooQCFQ5vOAodVR8rgg Accessed 23 March 2007
5. Shekhar, S., Chawla, S.: Spatial Databases: a Tour. Prentice Hall (2003)
6. Raster GIS http://www.gsd.harvard.edu/pbcote/courses/gsd6322/raster/index.htm Cambridge, MA, USA
7. ADHS Guides to Good Practice http://ahds.ac.uk/history/creating/guides/gis/ Accessed 21 March 2007

Raster Data Compression

► Data Compression for Network GIS

Raster Image

► Raster Data

Raster Models

► Data Models in Commercial GIS Systems

Rational Choice

► Crime Mapping and Analysis

Ra*-Tree

► Multi-Resolution Aggregate Tree

RCC

► Mereotopology

Realignment

► Conflation of Features

Real-Time Generalization

► Generalization, On-the-Fly

Real-Time Location Services

► Indoor Positioning

Reasoning

► Geospatial Semantic Web: Personalisation

Reasoning, Spatio-temporal

► Temporal GIS and Applications

Record-to-Record Travel Method

► Routing Vehicles, Algorithms

Rectangle, Hyper-

► Indexing, X-Tree

Rectangle, Minimum Bounding

► Indexing, X-Tree

Reference, Coordinate Based

► Indoor Localization

Reference, Symbolic

► Indoor Localization

Reference System, Spatial

► Geography Markup Language (GML)
► Oracle Spatial, Geometries

Reference System, Temporal

► Geography Markup Language (GML)

Reference-Feature Centric

► Co-location Patterns, Algorithms

Region, Conjeccture

► Vague Spatial Data Types

Region Connection Calculus

► Mereotopology

Regional Sience

► Time Geography

Regionalization, Spatial Ontologies

► Geodemographic Segmentation

Registration

JÉRÔME THÉAU
GIS Training and Research Center,
Idaho State University, Idaho, ID, USA

Synonyms

Coregistration; Co-registration; Georegistration

Definition

Registration is the process of geometrically matching different spatial datasets of the same area so that identical features are perfectly superimposed. In GIS, this match is usually performed within common geographic coordinate system. Spatial datasets may be from different sensors (e. g. satellite, aerial) and sources (e. g. maps, images), and acquired at different times (e. g. multidate satellite imagery).

Main Text

Registration is usually performed using one dataset as a reference (or master) and the other datasets as targets (or slaves) to correct. Registration can also be independently performed on different datasets (e. g. independent registration of multidate images to a common projection). In this case, registration does not involve any reference-target processing.

Basically, registration algorithms can be linear or nonlinear. Linear transformations involve translation, rotation, or scaling of the target dataset using, for example, first order transformations. Non-linear transformations are performed to correct for local and non-homogeneous deformations using, for example, polynomial transformations.

Registration algorithms are usually based on feature selection to perform the transformation, such as points, line intersections, or boundaries. However, some algorithms also use the structure of the image to perform the registration using, for example, Fourier analysis.

Registration is a critical step because many GIS applications such as change detection studies require two or more datasets of the same area. Accurate registration is therefore necessary to obtain accurate results.

Cross References

► Change Detection
► Co-location Pattern Discovery
► Correlation Queries in Spatial Time Series Data

Registry Information Model

► Catalogue Information Model

Regression

► Spatial and Geographically Weighted Regression

Regression, Geographically Weighted

► Geodemographic Segmentation

Reinsert, Forced

► R*-tree

Relative Location

► Localization, Cooperative

Relative Positional Accuracy

▶ Positional Accuracy Improvement (PAI)

Relevance, Spatial

▶ Retrieval Algorithms, Spatial

Relevance, Textual

▶ Retrieval Algorithms, Spatial

Reliable Real-Time Data Collection

▶ Data Collection, Reliable Real-Time

Remote Sensing

▶ Photogrammetric Sensors
▶ Standards, Critical Evaluation of Remote Sensing

Remote Sensing, Aerial

▶ Evolution of Earth Observation

Remote Sensing, Satellite-Based

▶ Evolution of Earth Observation

Remote Sensing Specifications

▶ Standards, Critical Evaluation of Remote Sensing

Remote Sensing Standards

▶ Standards, Critical Evaluation of Remote Sensing

Representational State Transfer Services

▶ Web Services

Representing Regions with Broad Boundaries

▶ Representing Regions with Indeterminate Boundaries

Representing Regions with Indeterminate Boundaries

ANTHONY G. COHN
School of Computing, University of Leeds, Leeds, UK

Synonyms

Egg-yolk Calculus; Representing regions with broad boundaries

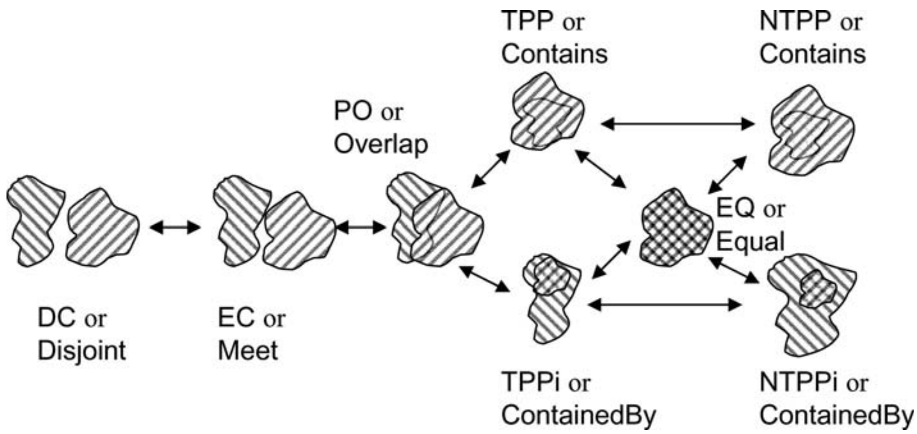
Definition

The problem of vagueness permeates almost every domain of knowledge representation. In the spatial domain, this is certainly true, for example it is often hard to determine a region's boundaries (e. g. "southern England"). Vagueness of spatial concepts can be distinguished from that associated with spatially situated objects and the regions they occupy. An adequate treatment of vagueness in spatial information needs to account for vague regions as well as vague relationships. Although there has been some philosophical debate concerning whether vague objects can exist,it is assumed here that they do, and some techniques for handling them are presented, specifically for considering the mereotopological relationships that may hold between such objects.

Historical Background

A number of approaches to representing and reasoning about regions with crisp boundaries had been developed by the mid 1990s [19,25] but the problem of treating regions with vague or indeterminate boundaries had not been specifically addressed. As a result of a workshop held in 1994 to investigate the problem of representing regions with indeterminate boundaries [7], two parallel but related calculi were proposed, each one based on one of the two main approaches to representing mereotopological relationships between two regions. The results [11,15] appear in in [7], and each approach has been further developed [12,16]. The approaches are also related to the notion of rough sets [6].

At least some of the same sorts of things about vague regions as about 'crisp' ones, with precise boundaries: that one contains another (southern England contains London, even if both are thought of as vague regions), that two overlap (the Sahara desert and West Africa), or that two are disjoint (the Sahara and Gobi deserts). In these cases, the two vague regions represent the space occupied by distinct entities, and we are interested in defining a vague area corresponding to the space occupied by either, by both, or by one but not the other. There might also be



Representing Regions with Indeterminate Boundaries, Figure 1 A 2D depiction of RCC-8 relations or the eight topological relations of the 4 and 9-intersection calculi. The arrows show the conceptual neighborhood structure

a requirement to say that one vague region is a 'crisper' version of another. For example, there might be an initial (vague) idea of the extent of a mineral deposit, and then information is received reducing the imprecision in the estimate of the deposit's extent. Here, the vagueness of the vague region is a matter of our ignorance: the entity concerned actually occupies a fairly well-defined region – though perhaps any entity's limits will be imprecise to some degree. In other cases, vagueness appears intrinsic: consider an informal geographical term like 'southern England'. The uncertainty about whether particular places (north of London but south of Birmingham) are included cannot be resolved definitively: it is a matter of interpretational context. A contrasting example is the region occupied by a cloud of gas from an industrial accident. Here are two sources of intrinsic vagueness: the concentration of the gas is likely to fall off gradually moving out of the cloud; and its extent will also vary over time, so any temporal vagueness (e. g., if asked about the cloud's extent 'around noon') will result in increased spatial vagueness.

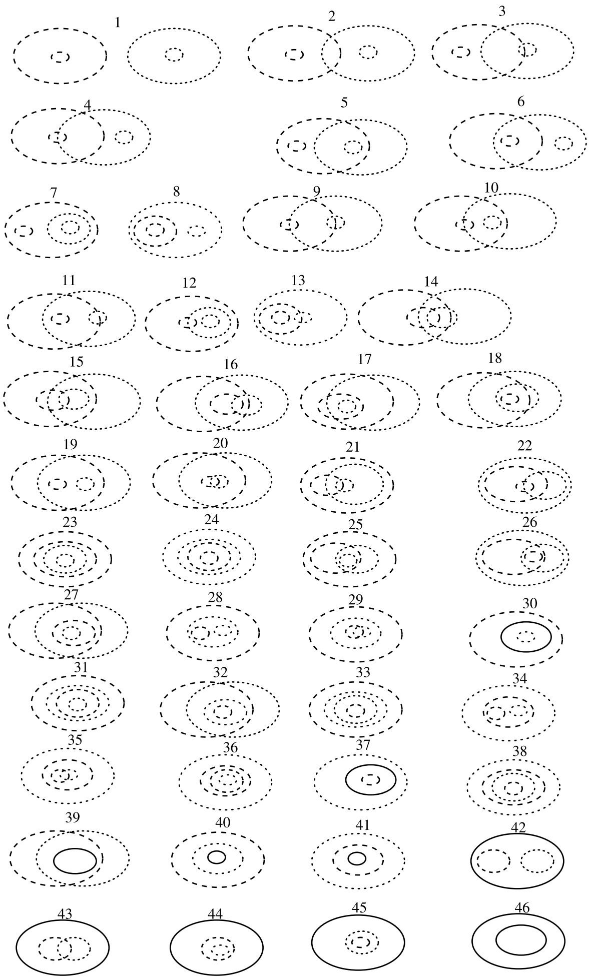
In these cases of intrinsic vagueness, there is a degree of arbitrariness about any particular choice of an exact boundary, and often, none is required. But *if* it is decided to define a more precise version (either completely precise, or less vague but still imprecise), our choice of version is by no means *wholly* arbitrary: it is possible distinguish more and less 'reasonable' choices of more precise description.

Distinguishing ignorance-based from intrinsic vagueness is important [2], but many of the same problems of representation and reasoning arise for both.

Scientific Fundamentals

There are two calculi in the literature which present sets of jointly exhaustive and pairwise disjoint mereotopological relations between pairs of regions. One is the so called Region Connection Calculus, which has two main variants, RCC-8 and RCC-5, with eight and respectively five *jointly exhaustive and pairwise disjoint* (JEPD) sets of relations. A 2D interpretation of the RCC-8 relations is depicted in Fig. 1; each of these relations can be defined within first order predicate calculus from a single primitive reflexive and symmetric relation **C**(x,y), true when region x is connected to region y. Alternatively, an essentially equivalent set of eight relations can be defined using the *4-intersection* method in which a relation between two regions x and y is defined in terms of whether the intersection of their boundaries and interiors is empty or non empty. An extension, the *9-intersection* method which also exploits the exterior of each region is also sometimes utilized. The RCC-5 calculus collapses the **DC** and **EC** relations to give a relation **DR**, the **TPP** and **NTPP** relations to give single proper part relation **PP** and similarly for **TPPi** and **NTPPi** giving **PPi**. The 4-intersection relations can be similarly reduced to five purely mereological relations.

These five relations have been extended to cover the case where the regions concerned do not have crisp boundaries. The extension of the RCC calculus [15,16] is called the "egg-yolk" calculus, while the extension of 9-intersection [11,12] is termed a spatial data model for objects with broad boundaries (henceforth SDMOBB). The egg-yolk calculus in fact originates in earlier work on data integration [23]. The approach in each of these is essentially the same – to identify a core region which always belongs to the region in question (the *yolk* in the terminology of former), and an extended region which might or might not be part of it (together forming the *egg*). There is a constraint that the yolk is always a proper part (**PP**) of the whole egg; the question then is: how many relations are there between the two regions, expressed in terms of how the respective yolks and eggs intersect? The answer is slightly different in the two cases, resulting from the two underlying construction techniques. In the egg-yolk case the RCC-5 relations are used to relate the components of each egg-yolk pair,



Representing Regions with Indeterminate Boundaries, Figure 2 The 46 relations of the egg-yolk calculus derived from RCC-5

and every consistent combination of RCC-5 relations over the 4 pairs of regions (egg_1-egg_2, egg_1-$yolk_2$, $yolk_1$-egg_2, $yolk_1$-$yolk_2$) is possible; it turns out there are 46 possible relations which are displayed in Fig. 2.

In the SDMOBB the 9-intersection method of [18] is modified so that the boundary becomes a broad boundary, and a set of constraints on consistent combinations of the 9-intersection are developed. This results in 44 relations. Analysis reveals that the two calculi are identical except that SDMOBB coalesces cases 30 and 31 of the egg-yolk calculus since the meeting of the broad boundaries in case 30 is regarded as an intersection just as is the case in 31 where there is a proper overlap. Similarly, cases 37 and 38 are coalesced in SDMOBB.

For any given configuration in the egg-yolk calculus one can consider the possible ways in which each region can be consistently "crisped"; depending on exactly how the crisp regions are formed (i. e. depending on exactly where the boundary between yolk and egg is drawn), different RCC-5 relations may pertain. However not all RCC-5 relations are possible for any given configuration. There are 13 equivalence classes formed from the 46 relations under this procedure, and these can be regarded as forming a coarser grained calculus for regions with indeterminate boundaries.

A similar clustering into groups of relations has also been achieved for the SDMOBB calculus, in this case into 14 clusters, by considering the possible geometric interpretations of each relation and clustering similar ones together. Four relations are excluded from this analysis in order to simply the graph – those cases where the broad boundary has to be large enough to encompass the other region entirely (relations 19, 28, 34, 42 using the egg-yolk numbering).

Conceptual neighbourhoods have been defined for RCC-5, RCC-8, the 9-intersection calculi and indeed many other qualitative spatial calculi, which specify which relations are conceptually "close"; this conceptual closeness is usually defined in the sense that continuous deformation or movement of the regions involved results in the change of relation to one of the immediately conceptually neighbouring ones. This notion can also be defined for the egg-yolk calculus and for SDMOBB.

The egg-yolk analysis can also be extended so that RCC-8 rather than RCC-5 is used as the underlying calculus. It turns out that if one generalizes RCC-8 in this way [16] there are 252 JEPD relations between non crisp regions which can be naturally clustered into 40 equivalence classes. An axiomatic presentation of the theory is also presented here which relies on an additional binary primitive relation to RCC (as well as the original $\mathbf{C}(x,y)$ relation) – x is crisper than region y. A mereolological framework which encompasses both standard RCC and the egg-yolk calculus in a semantic setting has also been developed [28].

It has been shown [12] that SDMOBB can be used to reason not just about regions with indeterminate boundaries but also can be specialized to cover a number of other kinds of regions including convex hulls of regions, minimum bounding rectangles, buffer zones and rasters. (This last specialization generalizes the application of the n-intersection model to rasters previously undertaken [20].) It is also noted that the calculus can be used for reasoning for regions with holes (by reinterpreting the "yolk" as a hole). A similar calculus for lines with broad boundaries has also been developed [10]. As an alternative to using the 9-intersection method, the "calculus based method" for defining topological relations [13] has also been applied to defining SDMOBB [14].

R

Another notion of indefiniteness relates to locations. Bittner [4] deals with the notion of exact, part and rough location for spatial objects. The exact location is the region of space taken up by the object. The notion of part location (as introduced by [8]) relates parts of a spatial object to parts of spatial regions. The rough location of a spatial object is characterized by the part location of spatial objects with respect to a set of regions of space that form regional partitions. Consequently, the notion of rough location links parts of spatial objects to parts of partition regions.

Bittner [4] argues that the observations and measurements of location in physical reality yield knowledge about rough location: a vaguely defined object *o* is located within a regional partition consisting of the three concentric regions: 'core', 'wide boundary' and 'exterior'. In this context, the notion of rough location within a partition consisting of the three concentric regions coincides with the notion of vague regions introduced by [15].

Key Applications

Of all application areas for reasoning about space, geography abounds in examples of objects whose precise extent or location is hard to pin down precisely; from hills/valleys, to meterological entities and habitats, few geographic entities can be precisely spatially located and/or bounded. Perhaps only political entities, with *fiat boundaries* [9] are precisely defined (even if sometimes in dispute!). The potential for applying theories which handle spatial indeterminacy is broad.

Future Directions

It is worth noting the similarity of these ideas to rough sets [17], though the exact relationship has yet to be fully explored, though see, for example [5,24]. Other approaches to spatial vagueness are through the use of supervaluation theory [2,3,22,26], the use of fuzzy calculi [7], to work with an indistinguishability relation which is not transitive and thus fails to generate equivalence classes [21,29] and the development of nonmonotonic spatial logics [1,27].

The issue of whether the egg-yolk calclulus (and equivalently SDMOBB) really capture the notion of spatial *vagueness* properly with a tripartite division of space has been discussed [16] and an argument presented that the egg-yolk calculus can provide a way of interpreting a more general, higher order, axiomatization of vagueness.

Cross References

► Conceptual Neighborhood
► Mereotopology

Recommended Reading

1. Asher, N., Lang, J.: When nonmonotonicity comes from distance. In: Nebel, B., Dreschler-Fischer, L. (eds.): KI-94: Advances in Artificial Intelligence. pp. 308–318. Springer-Verlag, Heidelberg (1994)
2. Bennett, B.: Application of supervaluation semantics to vaguely defined spatial concepts. In: Montello, D.R. (ed.) Spatial Information Theory: Foundations of Geographic Information Science; Proceedings of COSIT'01, volume 2205 of LNCS, pp. 108–123. Springer, Morro Bay (2001)
3. Bennett, B.: What is a forest? on the vagueness of certain geographic concepts. Topoi. **20**(2), 189–201 (2001)
4. Bittner, T.: On ontology and epistemology of rough location. In: Freksa, C., Mark, D.M. (eds.) Spatial Information Theory – Cognitive and Computational Foundations of Geographic Information Science, number 1661 in Lecture Notes in Computer Science, pp. 433–448. Springer, Proceedings of COSIT'99 (1999)
5. Bittner, T., Stell, J.G.: Rough sets in qualitative spatial reasoning. In: Rough Sets and Current Trends in Computing, Banff, October 2000, Proceedings, volume 2005 of Lecture Notes in Computer Science, pp. 445–453. Springer, Heidelberg (2001)
6. Bittner, T., Stell, J.G.: Stratified rough sets and vagueness. In: Kuhn, W., Worboys, M.F., Timpf, S. (eds.) Spatial Information Theory. COSIT'03 Proceedings, volume 2825 of Lecture Notes in Computer Science, pp. 286–303 (2003)
7. Burrough, P., Frank, A.M. (eds.): Specialist Meeting on Geographical Objects with Undetermined Boundaries, GISDATA. Francis Taylor (1996)
8. Casati, R., Varzi, A.: The structure of spatial localization. Philos. Stud. **82**, 205–239 (1996)
9. Casati, R., Varzi A.: Fiat objects. Topoi, **20**,131–148 (2001)
10. Clementini, E.: A model for lines with a broad boundary. In: 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2002),Annecy, France, number 692 in LNCS, pp. 1579–1586 (2002)
11. Clementini, E., Di Felice, P.: An algebraic model for spatial objects with undetermined boundaries. In: Burrough, P., Frank, A.M. (eds.): Proceedings, GISDATA Specialist Meeting on Geographical Entities with Undetermined Boundaries. Taylor Francis, London (1996)
12. Clementini, E., Di Felice, P.: Approximate topological relations. International Journal of Approximate Reasoning, **16**,173–204, (1997)
13. Clementini, E., Di Felice, P., Oosterom, P.: A small set of formal topological relationships suitable for end user interaction. In: Abel, D., Ooi, B.C. (eds.): Advances in Spatial Databases, Proceedings of the 3rd International Symposium on Spatial Databases (SSD'93), Singapore, number 692 in LNCS, pp. 277–295. Springer-Verlag, Heidelberg (1994)
14. Clementini, E., Di Felice, P.: A spatial model for complex objects with a broad boundary supporting queries on uncertain data. Data Knowl. Eng. **37**(3), 285–305, (2001)
15. Cohn, A.G., Gotts, N.M.: The 'egg-yolk' representation of regions with indeterminate boundaries. In: Burrough, P., Frank, A.M. (eds.): Proceedings, GISDATA Specialist Meeting on Geographical Objects with Undetermined Boundaries, pp. 171–187. Taylor Francis, London (1996)
16. Cohn, A.G., Gotts, N.M.: A mereological approach to representing spatial vagueness. In: Doyle, J. Aiello, L.C., Shapiro, S. (eds.): Principles of Knowledge Representation and Reason-

ing, Proc. 5th Conference, pp. 230–241. Morgan Kaufmann, San Mateo (1996)
17. Düntsch, I., Gediga, G.: Uncertainty measures of rough set predictions. Artif. Intell. **106**, 109–137 (1998)
18. Egenhofer, M.: Deriving the composition of binary topological relations. J. Vis. Lang. Comput. **5**(2), 133–149 (1994)
19. Egenhofer, M., Franzosa, R.: Point-set topological spatial relations. Int. J. of Geogr. Inf. Syst. **5**(2), 161–174 (1991)
20. Egenhofer, M.J., Sharma, J.: Topological relations between regions in R^2 and Z^2. In: David, A., Ooi, B. (eds.): Proceedings of the 3rd International Symposium on Advances in Spatial Databases, SSD'93, volume 692 of LNCS, Springer, Heidelberg (1993)
21. Kaufman, S.: A formal theory of spatial reasoning. In: Proc Int. Conf. on Knowledge Representation and Reasoning, pp. 347–356, Morgan Kaufmann, San Mateo (1991)
22. Kulik, L.: A geometric theory of vague boundaries based on supervaluation. In: Montello, D.R. (ed.): Spatial Information Theory: Foundations of Geographic Information Science, International Conference, COSIT 2001, Morro Bay, CA, USA, September 19–23, 2001, Proceedings, volume 2205 of Lecture Notes in Computer Science, pp. 44–59. Springer, Heidelberg (2001)
23. Lehmann, F., Cohn, A.G.: The EGG/YOLK reliability hierarchy: Semantic data integration using sorts with prototypes. In: Proc. Conf. on Information Knowledge Management, pp. 272–279. ACM Press, New York (1994)
24. Polkowski, L., Skowron, A.: Rough mereology in information systems. a case study: qualitative spatial reasoning. In: Rough set methods and applications: new developments in knowledge discovery in information systems, pp. 89–135. Physica-Verlag GmbH, Heidelberg, Germany, (2000)
25. Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. In: Proc. 3rd Int. Conf. on Knowledge Representation and Reasoning, pp. 165–176. Morgan Kaufmann, San Mateo (1992)
26. Santos, P., Bennett, B., Sakellariou, G.: Supervaluation semantics for an inland water feature ontology. In: Kaelbling, L.P., Saffiotti, A. (eds.): Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05), pp. 564–569. Professional Book Center, Denver (2005)
27. Shanahan, M.: Default reasoning about spatial occupancy. Artif. Intell. **74**(1),147–163 (1995)
28. Stell, J.G.: Part and complement: Fundamental concepts in spatial relations. Ann. Artif. Intell. Mat. **41**,1–18 (2004)
29. Topaloglou, T.: First order theories of approximate space. In: Anger, F. et al. (ed.): Working notes of AAAI workshop on spatial and temporal reasoning, Seattle, pp. 283–296 (1994)

Resolution-Based Matrix Quadtree

► Geospatial Authorizations, Efficient Enforcement

Resolving Semantic Schema Heterogeneity

► Database Schema Integration

Resource Description Framework (RDF)

► Geography Markup Language (GML)

Retrieval Algorithms, Spatial

Mark Sanderson
Department of Information Studies,
University of Sheffield, Sheffield, United Kingdom

Synonyms

Geographic information retrieval; Information retrieval; Toponym; Footprint; Synonymy; Ambiguity; Gazeteer; Triage, query; Relevance, spatial; Relevance, textual

Definition

Before describing means of adapting text search engines to deal with location, it is important to both define what is meant by spatial retrieval and further to establish how often users actually need access to search systems which provide a spatial component. Purves and Jones define spatial retrieval as [16]

"... the provision of facilities to retrieve and relevance rank documents or other resources from an unstructured or partially structured collection on the basis of queries specifying both theme and geographic scope."

Historical Background

In determining user needs for such location-based search, one can start by examining the range of existing search engines with explicit support for spatial search. Probably the most obvious starting point for examples of such systems is to examine the major Web search engines, such as Google, Yahoo or MSN. All provide so-called *local search* facilities, which essentially provide spatial search over yellow pages (i. e. directories of businesses). The sophistication of such searching is limited to querying for the name or type of businesses typically found in yellow pages and specifying a spatial footprint, using a postal code or the name of a population center. The prominence of local search on all three of the main engines indicates its prominence. It is also worth noting that search engines – when showing advertising next to their standard search results – try to provide adverts for goods and services located close to the queryier's actual location. This they do by examining the IP address of the searcher's computer, which can often be used to determine the location of the user at least to the spatial resolution of the country the searcher's computer is in [5].

R

For more general search of web pages, it would appear that the search engines do not provide any explicit support for spatial search[1]. In analyses of query logs, it was found that around 20% of searches included a location within them [13]. The analyses examined the range of spatial relations that users specified in their searches, of which almost all used "in", others such as "near", "north of", "surrounding", etc. were much less common, (occurring one or two orders of magnitude less frequently). A simple word based search for a particular topic (e. g. battles) *in* a particular location (e. g. some region of a country) is well served by the search engines' usual requirement that matching documents contain all words of the users' query. Even though other spatial relations aren't as well served by conventional text search, because they aren't used as much, it would appear search engines aren't motivated to alter their engines to support the relations.

From such an examination of services provided and the queries submitted to search engines, it would appear that users conduct spatial search relatively often, but it would also appear that, in general, insufficient numbers of users require support for complex spatial relations to make is worthwhile providing.

Spatial Search Needed on the Web?

A further reason that Web search engines may not be providing specific support for spatial search is that even when users search for objects with a particular spatial region that isn't mentioned on a particular web page, often such pages are still easy to find. There are a great many Web pages that simply list items found within a particular spatial area and provide pointers to the page describing those items in more detail. A series of searches on any Web search engine will confirm the presence of such lists: Castles in Scotland, hotels in New England, for example. Even if the Web site of a hotel in downtown Boston fails to mention that the hotel is in New England, the page will still be findable via a list page. Even searches for apparently obscure items such old yew trees in England or Vitrified Forts in Yorkshire often produce relevant useful list pages, which satisfy most searchers. (Academic research on geographic text search has confirmed the success of simple keyword search when searching for locations [6]. With this combination of straightforward user queries coupled with large sets of list pages grouping items spatially, it is perhaps unsurprising that search engines appear to do little to further support spatial search.

[1] It should be noted that the exact means by which Web search engines provide their services are closely guarded secrets and the methods used by them change continuously. Therefore, any statement about how such search engines work (from any source, not just this text) should be taken to be little more that informed speculation that might go out of date at any time.

Key Applications

It would be wrong to assume that the services offered by the major Web search engines in some way cover or define all possible searching needs. Users searching for other types of objects or searching in other domains can have quite different priorities. One example of this is searching for photographs via text captions. Here, location appears to be of great importance. An examination of requests made by journalists to photo libraries [3], and more recently of query logs maintained by image digital libraries [12] revealed that between a 1/3 and 1/2 of all requests to such libraries were related to a specific place, region or country; or the request was for a certain form of landscape (e. g. river, sea, countryside, mountains); or a particular permanent man made feature that would typically be represented in a GIS, (e. g. railway station, school, park, church, etc.). An examination of the keywords manually assigned to images in Flickr reveals a similar emphasis on location.

There would also appear to be an increase in the number of cameras produced that provide some form of location information either through GPS or on camera phones that determine location using data from the cell phone operators [1]. From such devices one might be able to automatically caption photographs and research is on-going to examine the potential of this approach [11,7].

Specialized Spatial Search Services

It would also appear that there are specialized search problems that require consideration of spatial search. For such problems, companies exist to provide solutions. At the time of writing probably the best example of such a company is MetaCarta, which provides spatial search tools for such needs.

From this overview, it is clear that spatial search is a common though not dominant form of search and that different forms of searching require different types of spatial search. The means one can use to support such search are next described.

Scientific Fundamentals

Spatial search requires different components in order for it to be supported well. First, it may be necessary to determine if users are issuing a spatial search, if they are, then both the user's query and each of the documents in the collection to be searched need to have some form of location associated with them. When a search to such a system is issued, those documents that are both semantically and spatially relevant to the user's query need to be found as quickly as possible. This Section describes these stages of spatial search, starting with a brief overview of the basic task of determining if a text contains a reference to a location (formally known as a *toponym*) within it and establishing what area the location is referring to (known as a *footprint*).

Identifying Toponyms in Text

A detailed exploration of toponym identification and resolution is beyond the scope of this Section, however, accurately spotting all toponyms and resolving which location is being referred to is not as straightforward as one might initially imagine. As with language in general, there are two basic problems in identifying toponyms and resolving (*grounding*) the toponyms to a location: *synonymy* (different ways of referring to the same thing) and *ambiguity* (a word having different meanings). Each is briefly described here.

Synonymy

There are many different ways in which a location can be referred to in a text and these ways can be both explicit, implicit and vague. The commonest explicit way of referring to a location in text is simply to provide the name of a location. Identifying such a location in a text can simply be achieved with the use of a large place name gazetteer, of which both open source and licensed versions can be found. A full address in a text is another common example of an explicit location and again gazetteers containing lists of addresses have been created.

One can also implicitly determine location by examining items such as telephone numbers. As with names of places and addresses, gazetteers that list what location corresponds to a particular phone number exist. There can also be domain specific implicit references to location.

- Wang showed that many user Web queries can be implicitly spatial [17]: one word queries for services, such as cinemas or pizza restaurants are likely to be best served by a spatial search if the search engines has access, through some other means, to the location of the user issuing the query.
- If a text comes to a computer associated with an IP address (for example query text transmitted to a remote search engine), most IP addresses can be resolved to an accuracy of at least a country, often to a city and sometimes to very precise locations (e. g. if an IP address comes from a free wireless service at an Internet café).
- If one is trying to associate a location with a set of Web pages, one might identify a location on one page, e. g. the "Contact Us" page of a business Web site and the assume other pages from the same Web site have the same location associated with them.

An often overlooked form of reference to locations are *imprecise regions*: these are vague locations commonly used whose boundaries aren't precisely defined. Examples of such areas are

- the "Midlands" of England a large area centered on the city of Birmingham;
- the "Eastern Seaboard" of the United States, a long region of the US east coast encompassing cities such as Boston, New York, Washington, etc., but whose western boundaries aren't as clearly defined;
- the "Red Centre" of Australia, the largely un-inhabited central dessert region of the country.

Resources to help determine the location and shape of such regions are sparse, despite use of such regions being relatively common in both documents and in user queries. Research on automatically establishing the shape of imprecise regions using text mining approaches has been reported [2].

Ambiguity

In general, words are ambiguous, with many words having more than one meaning: the classic example of such ambiguity is the word "bank", which can refer to both an economic institution and the side of river, as well as other less well known meanings. Unfortunately, such ambiguity extends to toponyms as well. This has been studied by Smith and Mann [14], who showed that particularly in countries colonized relatively recently, such as those in North and Central America, many place names are re-used. For example, there is a "Springfield" in almost every state of the US. Unfortunately, place names are not necessarily just spatially ambiguous, they can also be ambiguous with other types of words. They could also be:

- names of people (e. g. the state of "Victoria" in Australia; the city of "Washington" in the US, etc.);
- used by organizations taking on the name of their location (e. g. football clubs, such as "Manchester United", "Real Madrid", "AC Milan", etc.);
- used metaphorically (e. g. one might use the term "Hollywood" to refer to the US film industry rather than the place);
- used in different languages having quite different meanings (e. g. there is a place in Germany called "the");
- used to mean different things (e. g. a 5 digit US postal code might simply be used in a text as a number).

Accurately identifying words as toponyms has been studied for many years with research conducted as part of the Message Understanding Conferences (MUC) of the late 1990s. More recently, work on toponym resolution has increased. Research and an extensive literature review from Leidner showed that fully automated accurate toponym resolution is hard to achieve and is still an active area of research [9].

Query Triage

Depending on the form of search being conducted, it may be necessary to automatically determine if a user's query is a spatial one or not. Examples of systems that need to make such a determination are Web search engines, which by default search for web pages, but may need to direct a spatial query to a local search service. Some work in this area of so-called *query triage* has been published [17]. In most cases, simply examining the user's query for the presence of a place name (taken from a large gazetteer) is sufficient for identifying a potentially spatial query.
Once a spatial search has been identified, the next stage is to establish the components of the query: the subject of the query, the location the query pertains to and possibly, the spatial relation between the subject and the query. Recognizing the spatial relation in a query text is a relatively un-explored aspect of research, however a simple search of words commonly used to define a spatial relation (e. g. "in", "north of", "near", "close to", etc.) is likely to suffice most times.
A much simpler way of determining the components of a spatial query is to design the user interface of a search system to capture these elements individually.

Conducting a Spatial Search

Assuming one has received a query with an identified location and relation that will search a collection of documents which themselves each have a location associated with them, one needs to consider how best to retrieve the documents. When matching documents to a spatial query, there are two aspects to be considered: the textual relevance of a document to the query as well as the document's spatial relevance.

Textual Relevance

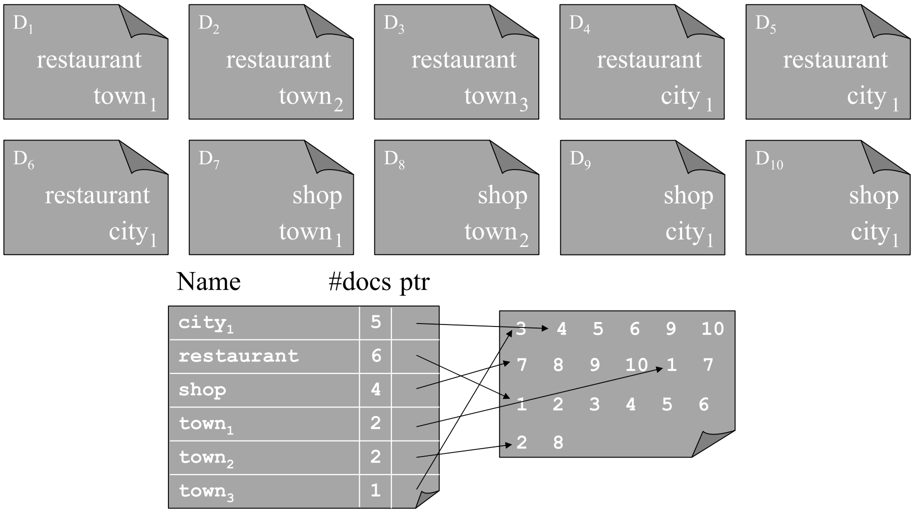
Details on the means to determine textual relevance is beyond the scope of this document. The exact methods used to calculate this very much depend on the type of documents being searched: for document collections, such as office files, newspaper articles or email finding, documents holding the user's query words repeated many times within the document is a simple but effective approach (see [4] for more coverage on this topic). If searching Web documents, consideration of the number of links pointing to a page and the *anchor text* written in those links takes priority over query word density (see [10] for more details).

Spatial Relevance

Determining the spatial similarity of a document to a query starts by establishing the spatial footprint of the query, followed by selecting documents that are relevant to the spatial area of the query. The footprint of a query is determined by the location and any spatial relation specified in the query. Determining the size and shape of a footprint is dependent on the subject of the query: if users are searching for restaurants in a city, they are likely to be implying that the eateries must be found within the city limits; however users searching for an airport in a city, may be happy with matches well beyond the city's border.
Once a footprint has been established, documents found to be both textually and spatially relevant are retrieved. Means of combining the scores of the spatial and textual components were described by van Kreveld et al. [8]. One important aspect to searching spatially is ensuring that the documents returned to users are not just those spatially relevant a particular sub-area of the query footprint, but are instead dispersed as evenly as possible around the footprint area. The van Kreveld paper described algorithms for achieving this dispersion searching and reported improved user satisfaction with such a searching system.

Internals of Search

Speed of search is a crucially important aspect of search. Studies have found that users use search engines less frequently if the speed of response of the engine is anything more than a 1/4 of a second. Ensuring that search is completed a quickly as possible is critical to building a successful searching system. In this Section an overview of the basic method used by ordinary text search engines to provide fast search is described, followed by a description of the means used to adapt a search engine to serve spatial queries.
To achieve fast search over a large collection of documents, a standard search engine uses a combination of a sorted word *index* and an *inverted file* (sometimes referred to as a *postings file*) to facilitate fast search of documents matching a particular query. Figure 1 shows a simplified example of such a combination of files, built from a collection of ten "toy" documents, which describe a set of shops and restaurants in three towns (named $town_1$, $town_2$, $town_3$ and one city (named $city_1$). Each document is identified by an ID number ranging from 1–10. The index holds a sorted list of all the words contained in all the documents. Each entry in the index has the word, the number of documents that word occurs in, and a pointer to the place in the inverted file where the ids of the documents holding that word are located. For a detailed outline of how such files are created and means of refining the structures shown here, see [18].

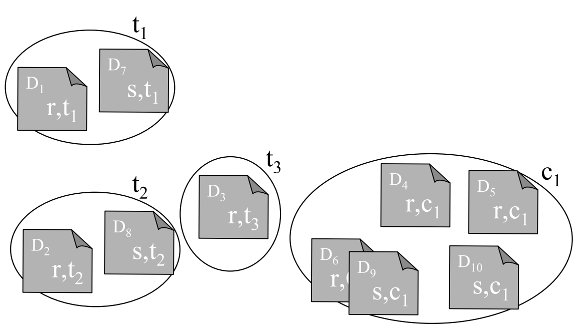


Retrieval Algorithms, Spatial, Figure 1 'Toy' collection and a conventional word-based index of the collection

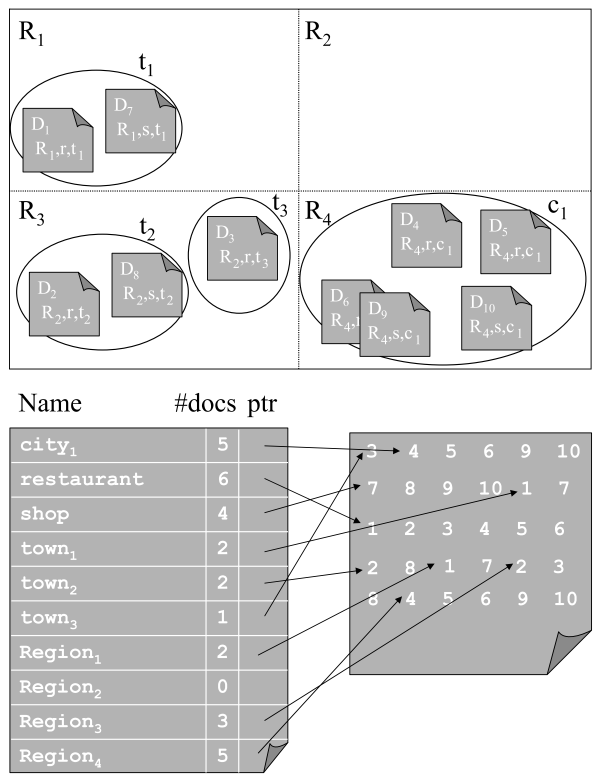
Adapting Text Search to Spatial Search

Such an index already supports simple spatial search: users wishing to find a restaurant in $town_1$ simply type the query "restaurant $town_1$"; Taking the intersection of the list of document IDs containing the word restaurant and the list containing the word $town_1$ results in a single document, which is shown to the user. If users wished to find restaurants near to and in $town_1$, it would be necessary to know the spatial location of each town and city. With such information (see Fig. 2), if it was judged that $town_2$ and $town_3$ were near to $town_1$, a spatial search system could simply issue three separate queries "restaurant $town_1$", "restaurant $town_2$", "restaurant $town_3$" and show the union of the returned results.

This simplistic approach of conducting multiple searches, one for each location that is spatially relevant to a query is a reasonable method if the number of named locations is relatively small, as is the case with the example collection. If the collection is much larger, however, issuing multiple queries starts to be time consuming. Borrowing methods



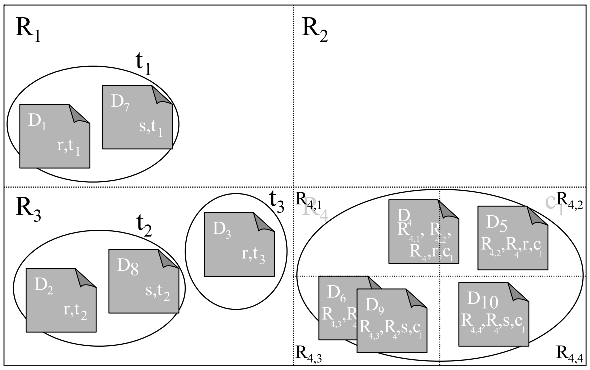
Retrieval Algorithms, Spatial, Figure 2 'Toy' collection spatially arranged



Retrieval Algorithms, Spatial, Figure 3 Organising documents into regions, adding region keywords to the index

from spatial indexing, the map in Fig. 3 can be broken into regions (R_1, R_2, R_3, R_4) and keywords[2] associated with each of these regions can be assigned to the relevant documents (see Fig. 4).

[2] Care needs to be taken to ensure that such keywords do not clash with actual words found in documents.



Retrieval Algorithms, Spatial, Figure 4 Further sub-dividing one region into smaller spatial areas

If a particular area, such as the city, isn't spatially represented in sufficient detail, its region, R_4, can be further sub-divided ($R_{4,1}$, $R_{4,2}$, $R_{4,3}$, $R_{4,4}$) and keywords associated with each region additionally assigned to the five documents in that region. If a document is found to be spatially relevant to more than one region, it can be assigned multiple region keywords. With such a representation in place, queries for shops in the south west of the city can be issued with a manageable number of searches, as can queries for restaurants near to $town_1$. Other types of data structure for providing fast spatial search have been examined and are described in [15], which also contains an extensive review of other spatial search research.

Future Directions

With the rapid uptake of geo-location devices, such as GPS, being placed in more mobile devices, such as cell phones and digital cameras, it is reasonable to expect that search over data that has a location associated with it will increase and the consequent important of spatial retrieval algorithms will grow.

Cross References

► Indexing Spatio-temporal Archives
► Representing Regions with Indeterminate Boundaries

Recommended Reading

1. Ahern, S., Davis, M., King, S., Naaman, M., Nair, R.: Reliable, User-Contributed GSM Cell-Tower Positioning Using Context-Aware Photos. In: The Eighth International Conference on Ubiquitous Computing (UbiComp) (2006)
2. Arampatzis, A., van Kreveld, M., Reinbacher, I., Jones, C.B., Vaid, S., Clough, P.D., Joho, H., Sanderson, M.: Web-based delineation of imprecise regions. J. Comput. Environ. Urban Syst. (CEUS) **30**(4) (2006)
3. Armitage, L.H., Enser, P.G.B.: Analysis of user need in image archives. J. Inf. Sci. **23**(4): 287–299 (1997)
4. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval, Addison Wesley, (1999) ISBN-13: 978–0201398298
5. Buyukkokten, O., Cho, J., Garcia-Molina, H., Gravano, L., Shivakumar, N.: Exploiting geographical location information of web pages. In: Proceedings of Workshop on Web Databases (WebDB'99) held in conjunction with ACM SIGMOD'99 (1999)
6. Gey, F., Larson, R., Sanderson, M., Joho, H., Clough, P.: GeoCLEF: the CLEF 2005 Cross-Language Geographic Information Retrieval Track. In: The Working Notes for the CLEF 2005 Workshop (2005)
7. Iwasaki, K., Yamazawa, K., Yokoya, N.: An Indexing System for Photos Based on Shooting Position and Orientation with Geographic Database. In: Proceedings of IEEE international conference on multimedia and expo (ICME), 390–393 (2005)
8. van Kreveld, M., Reinbacher, I., Arampatzis, A., van Zwol, R.: Distributed Ranking Methods for Geographic Information Retrieval. In: Developments in Spatial Data Handling, pp. 231–243. Springer, Berlin, Heidelberg (2004)
9. Leidner, J.L.: Toponym Resolution in Text: Annotation, Evaluation and Applications of Spatial Grounding of Place Names. PhD thesis, School of Informatics, University of Edinburgh, Edinburgh, Scotland, UK (2007)
10. Levene, M.: An Introduction to Search Engines and Web Navigation. Pearson, (2005) ISBN13: 9780321306777
11. Naaman, M., Song, Y.J., Paepcke A., Garcia-Molina H.: Assigning Textual Names to Sets of Geographic Coordinates. J. Comput. Environ. Urban Syst. J. **30**(4):418–435 (2006)
12. Paterson, C.: The effectiveness of using machine translators to translate German and Portuguese into English when searching for images with English captions. Masters Dissertation, Department of Information Studies, University of Sheffield (2002)
13. Sanderson, M., Kohler, J.: Analyzing geographic queries. In: The proceedings of Workshop on Geographic Information Retrieval SIGIR (2004)
14. Smith, D.A., Mann G.S.: Bootstrapping toponym classifiers. In: Proceedings of the Workshop on the Analysis of Geographic References, at the NAACL 2003 Conference (2003)
15. Purves, R.S., Clough, P., Jones, C.B., Arampatzis, A., Bucher, B., Finch, D., Fu, G., Joho, H., Syed, A.K. Vaid, S., Yang, B.: The design and implementation of SPIRIT: a spatially aware search engine for information retrieval on the Internet. Int. J. Geogr. Inf. Sci. (2007)
16. Purves, R.S., Jones, C.B.: Geographic Information Retrieval (GIR) editorial. In: special issue Comput. Environ. Urban Syst. **30**(4): 375–377 (2006)
17. Wang, L., Wang, C., Xie, X., Forman, J., Lu, Y., Ma, W., Li, Y.: Detecting Dominant Locations from Search Queries. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005) (2005)
18. Zobel, J., Moffat, A.: Inverted Files for Text Search Engines. ACM Computing Surveys, **38**(2): 1–56 (2006)

Reuse

► Smallworld Software Suite

Reverse-K-Nearest-Neighbors aNN

▶ Nearest Neighbors Problem

Reverse-Nearest-Neighbor-Problem

▶ Nearest Neighbors Problem

Reversible and Convertible Lanes

▶ Contraflow for Evacuation Traffic Management

Revisit Period

▶ Temporal Resolution

RF Identification

▶ Radio Frequency Identification (RFID)

RFID

▶ Radio Frequency Identification (RFID)

Rich Client Internet Applications

▶ Scalable Vector Graphics (SVG)

rkNN

▶ Nearest Neighbors Problem

RMS Error

▶ Photogrammetric Products

rNN

▶ Nearest Neighbors Problem

Road Maps

▶ Road Maps, Digital

Road Maps, Digital

BETSY GEORGE, SHASHI SHEKHAR
Department of Computer Science and Engineering,
University of Minnesota, Minneapolis, MN, USA

Synonyms

Digital road networks; Road maps

Definition

A digital road map is a representation of a physical road network that can be displayed or analyzed by a digital computer. Figure 1 shows a road map and its graph representation. Road intersections are often modeled as vertices and the road segments are connecting adjacent intersections represented as edges in the graph. For example, the intersection of 'SE 5th Ave' and 'SE University Ave' is modeled as node N1. The segment of 'SE 5th Ave' between 'SE University Ave' and 'SE 4th Street' is represented by the edge N1-N4. The directions on the edges indicate the permitted traffic directions on the road segments.
Digital road maps have gained importance due to the widespread use of location-based services such as route finding. They are essential in any location-based utility that involves route-based queries.

Historical Background

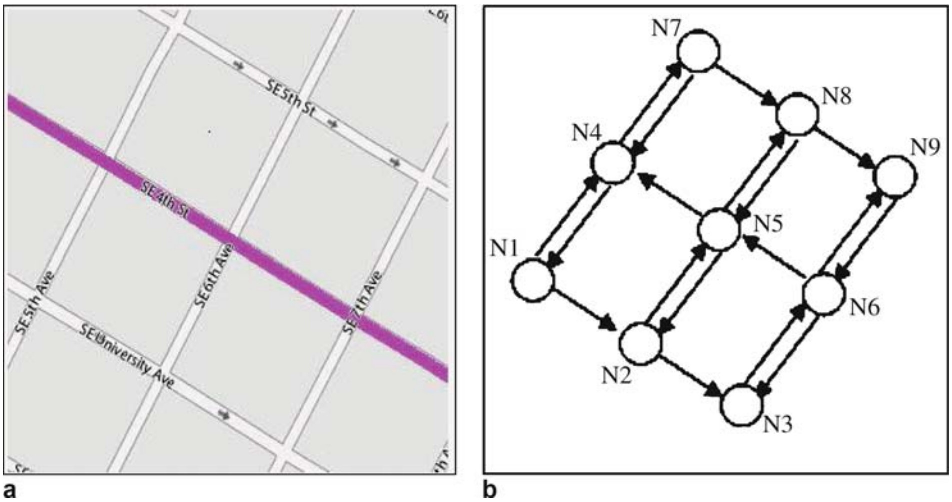
Traditionally, governmental agencies (e. g. state departments of transportation) were the primary sources of digital road maps. The initial efforts to formulate digital maps were undertaken by governmental agencies (e. g. state departments of transportation) through the digitization of paper maps from various sources. TIGER (Topologically Integrated Geographic Encoding and Referencing system) was conceived and developed in the 1980's in preparation for the 1990 Census by the US Census Bureau. Due to an increased popular demand, several private-sector companies have begun to offer digital road maps appended with additional points of interest and improve map accuracy.

Scientific Fundamentals

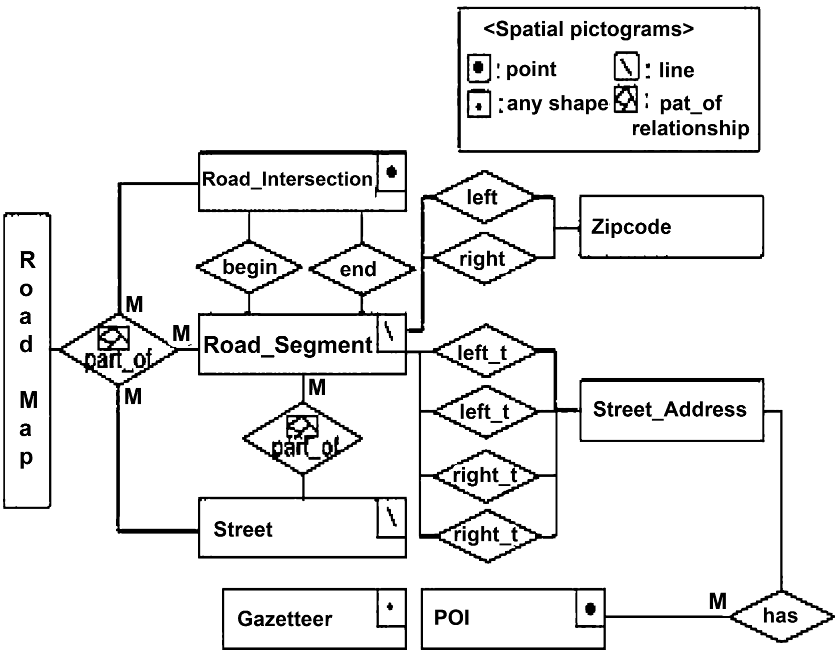
Data Model of Digital Road Maps

As in any database application, data modeling of digital road maps is critically important. The database design involves three steps namely conceptual modeling, logical modeling and physical modeling [2].

Conceptual Data Model The purpose of conceptual modeling is to adequately represent the data types, their



Road Maps, Digital, Figure 1 Road map and its graph representation (Source for Figure 1a: www.maps.yahoo.com)



Road Maps, Digital, Figure 2 A PEER diagram for a digital road map [1]

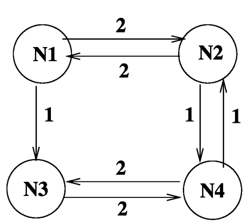
relationships and the associated constraints. Entity Relationship (ER) model, that has been widely used in conceptual modeling does not offer adequate features to capture the spatial semantics of road maps. Several extensions to address this limitation have been proposed. The pictogram enhanced ER (PEER) model [3] models road maps as graphs. Figure 2 shows a PEER diagram for a digital road map. In a digital road map, vertices represent road intersections and edges represent road segments. Labels and weights can be attached to vertices and edges to encode additional information such as names and travel times. A road segment is modeled using a range of street addresses, which are divided into left-side and right-side addresses. Each side of the road segment is represented using the two end-addresses. The zip code information of a street address can be used when the exact address is not known. Zip codes corresponding to the left and right sides of the streets are also stored. Two edges are considered to be adjacent if they share a common vertex. A PEER diagram can also represent points of interest.

Logical Data Model In the logical modeling phase, the conceptual data model is implemented using a commercial database management system. Among the various implementation models such as hierarchical, network, relational, object-relational data models and object-oriented models, the object-relational model has been gaining popularity in the representation of spatial applications. This model is supported by SQL3 and provides a mechanism for user-defined data types, thus allowing the definition of user defined complex data types such as point, line and polygon. [3] provides the grammar-based translation scheme to translate the PEER model into an object relational (OGIS/SQL3) model. In general, entity pictograms translate into appropriate data types in SQL3 and the relationship pictograms translate into spatial integrity constraints [1].

Physical Data Model The physical data modeling phase deals with the actual implementation of the database application. Issues related to storage, indexing and memory management are addressed in this phase. Very often, queries that are posed on a network database such as a road map, involve route finding. This means the database must provide adequate support for network computations such as finding shortest paths. Figure 3 shows three representations of a graph. Adjacency-matrix and adjacency list are two well-known data structures used for implementing road networks [4], represented as graphs. In an adjacency-matrix, the rows and columns of a matrix represent the vertices of the graph. A matrix entry can be either 1 or 0, depending on whether there is an edge between the two vertices as shown in Fig. 3b. An adjacency list (shown in Fig. 3c) consists of an array of pointers. Each element of the array represents a vertex in the graph and the pointer points to a list of vertices that are adjacent to the vertex. Directed graphs can be implemented in the relational model using a pair of relations, one for the nodes and the other for the edges. The 'Node' (R) and the 'Edge' (S) relations are shown in Fig. 3d and a denormalized representation is shown in Fig. 3e. The denormalized representation of a node table contains the coordinates of the node, a list of its successors and a list of its predecessors. This representation is often used in shortest path computations.

A spatial access method called the Connectivity-Clustered Access Method (CCAM) was proposed in [6], which clusters the vertices of the graph based on graph partitions, thus providing grouping of records into disk pages based on connectivity.

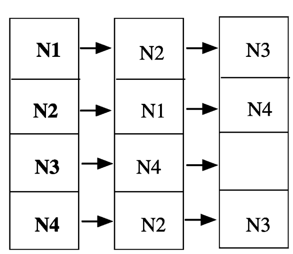
Turn Restrictions Turn restrictions are frequently encountered in road networks and they can affect the traversal in the network. A physical model that does not consider turn restrictions can lead to the computation of routes that are not entirely feasible. Turns have been modeled using a turn table where each turn restriction is represented as a row in the table that references the two associated edges [9]. Another proposed method to represent turn restrictions is node expansion [10]. The node that corresponds to a junction is expanded to a subgraph where permissible turns are represented as edges. This technique can lead to a substantial increase in the size of the network which adversely affects the performance. Another method involves the transformation of the road network to a line graph where the edges in the original network are mapped to vertices in the line graph and the turns are represented as edges in the line graph [9,11]. A representation that



a Example network graph

	N1	N2	N3	N4
N1	0	1	1	0
N2	1	0	0	1
N3	0	0	0	1
N4	0	1	1	0

b Adjacency matrix



c Adjacency list

Node (R)

ID	X	Y
N1	1	1
N2	3	1
N3	1	0
N4	3	0

Edge(S)

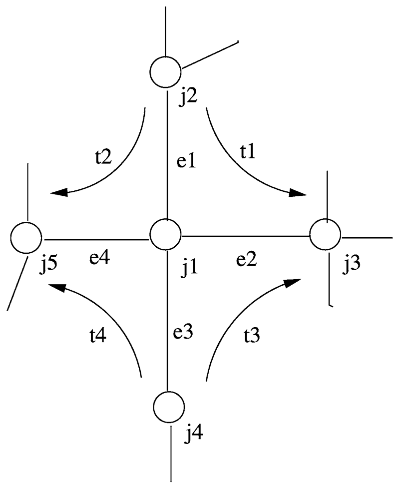
Src	Dst	Dist
N1	N2	2
N1	N3	1
N2	N1	2
N2	N4	1
N3	N4	2
N4	N2	1
N4	N3	2

d Node and Edge Relations

ID	X	Y	Succ	Pred
N1	1	1	(N2,N3)	N2
N2	3	1	(N1,N4)	(N1,N4)
N3	1	0	N4	(N2,N3)
N4	3	0	(N2,N3)	(N2,N3)

e Denormalized Node Table

Road Maps, Digital, Figure 3 Three different representations of a graph



a An example network

id	from–jn
e1	j2
e2	j3
e3	j4
e4	j5

b Edge Table

id	edge1	junc1	edge2	junc2	edge3	junc3	edge4	junc4
j1	e1	j2	e2	j3	e3	j4	e4	j5

c Junction Table

id	Turn id	First edge id	Last edge id	Turn id	First edge id	Last edge id	Turn id	First edge id	Last edge id	....
j1	t1	e1	e2	t2	e1	e4	t3	e3	e2	

d Turn Table

Road Maps, Digital, Figure 4 Representation of Turn Restrictions (adapted from [5])

consists of a junction table, edge table and turn tables was proposed in [5].

Every junction is represented as a row in the junction table. A row corresponding to a junction stores the edges that converge at the junction and the junctions connected to the given junction. The edge table stores edge identifiers and the junction where the edge originates (from-junction). A tuple in the turn table corresponds to a junction in the network. Each tuple consists of a junction identifier, and a triplet (turn identifier, first edge-id, last edge-id) corresponding to each turn associated with the given junction.

Figure 4 illustrates the representation of turn restrictions in a road network. Figure 4a shows a part of a road network around a junction *j1* where the edges *e1, e2, e3* and *e4* meet. The curved arrows indicate the permitted turns at the junction. For example, a turn is allowed from edge *e1* to edge *e*2. Figure 4b, c and d show the edge, junction and turn tables respectively, corresponding to turn *t1* in the example network. The junction table lists the edges that converge at junction *j1* (*e1, e2, e3,* and *e4* and the junctions connected to it (*j2, j3, j4,* and *j5*). The turn able shows the permitted turns at junction *j1* and the edges that participate in each turn. For example, turn *t1* represents a turn from edge *e1* to edge *e2* as illustrated by the *'first edge id'* and *'last edge id'* entries in the turn table in Fig. 4d.

Data Quality

Given the significant number of sources for the road map data and the heterogeneity across the data, it became necessary to define data quality in the context of digital road maps. Data quality refers to the relative accuracy and precision of a particular road map database. The purpose of the data quality report is to provide adequate information to the users to evaluate the fitness of the data for a specific use. There are several map accuracy standards, including the well-known National Map Accuracy Standard (NMAS) and the American Society for Photogrammetry and Remote Sensing (ASPRS) standard [1]. The standards consist of four components namely:

1. Lineage: This component deals with the narrative of the source materials used and procedures adopted to build the product.
2. Positional Accuracy: This defines the error in position of features. In digital road maps, this component is the most critical.
3. Attribute Accuracy: This represents the expected error in attributes such as road names.
4. Completeness: This defines the fraction of the real-world features represented on a map.

In addition, topological consistency is of concern for digital road maps in the context of navigation systems to facilitate graph computations such as shortest path algorithms.

Key Applications

Location-based Services

Digital road maps are indispensable for any location-based service that involves position or route based queries. Location-based services (LBS) provide the ability to find the geographical location of a mobile device and subsequently provide services based on that location. A digital road map is a key component of spatial database servers that provide efficient query-processing capabilities such as finding the nearest facility (e. g. a restaurant) and the shortest path to the destination from a given location [14,15]. Route-finding queries typically deal with route choice (shortest route to a given destination), destination choice (the nearest facility from the given location) and departure time choices (the time to start the journey to a destination so that the travel time is minimized). Though a significant amount of work has been done to find best routes and destinations, the problem of computing the best time to travel on a given route (time choice) needs further exploration.

Digital road maps are critical in in-vehicle navigations systems [12,13] where the maps would be used to compute the required routes on user-demand or to find points of interest.

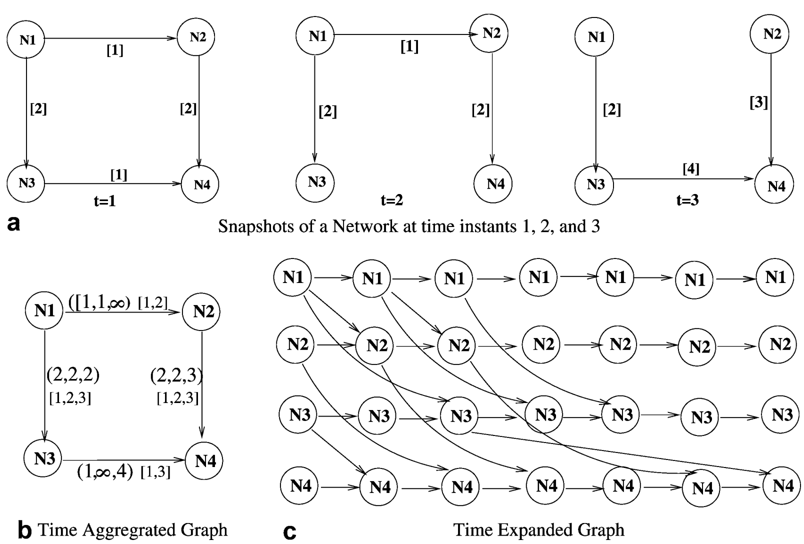
Emergency Planning

One key step in emergency planning is to find routes in a road network to evacuate people from disaster-stricken areas to safe locations in the least possible time. This requires finding shortest routes from disaster areas to destinations. In metropolitan-sized transportation networks, manual computation of the required routes is almost impossible, making digital road maps an integral part in the efficient computation of these routes.

Future Directions

A significant fraction of queries that are posed on a road network involves finding the shortest path between a pair of locations. Travel times on the road segments, very often depend on the time of day due to varying levels of congestion, thus making the shortest paths also time-dependent. Road networks need to be modeled as spatio-temporal networks to account for this time-dependence. Various models such as time-expanded networks [7] and time-aggregated graphs [8] are being explored in this context. Time expanded graph represents the time-dependence by copying the network for every time instant whereas in time aggregated graphs, the time-varying attributes are aggregated over edges and nodes.

Figure 5(a) shows a network at three time instants. The network topology and parameters change over time. For example, the edge N3-N4 is present at time instants $t = 1, 3$ and disappears at $t = 2$ and its weight changes from 1 at $t = 1$ to 4 at $t = 3$. The time aggregated graph that represents this dynamic network is shown in Fig. 5(b). In this figure, the edge N3-N4 has two attributes, both time series. The attribute [1,3] represents the time instants at which the edge is present and (1,∞,4) is the weight time series, indicating the weights at various instants of time. Figure 5(c) shows the time expanded graph that represents the same scenario. Edge weights in a time expanded graph are not explicitly shown as edge attributes; instead they are



Road Maps, Digital, Figure 5 Two different representations of a time-variant network

R

represented by edges that connect the copies of the nodes at various time instants. For example, the weight 1 of edge N1-N2 at $t = 1$ is represented by connecting the copy of node N1 at $t = 1$ to the copy of the node N2 at time $t = 2$. The time expansion for the example network needs to go through 7 steps since the latest time instant would end in the network is at $t = 7$. For example, the traversal of edge N3-N4 that starts at $t = 3$ ends at $t = 7$, the travel time of the edge being 4 units.

Cross References

► Contraflow in Transportation Network
► Emergency Evacuations, Transportation Networks
► Nearest Neighbor Queries in Network Databases

Recommended Reading

1. Shekhar, S., Vatsavai, R., Ma, X., Yoo, J.: Navigation Systems: A Spatial Database Perspective. In: Schiller, J. and Voisard, A. (eds.) Location-Based Services, Chapter 3. Morgan Kaufmann, San Francisco, CA, USA, April (2004) ISBN-13: 978-1558609297
2. Elmasri, R., Navathe, S.B.: Fundamentals of Database Systems (5th edn.), Addison Wesley, Boston, MA, USA, March (2006) ISBN-13: 978-0321369574
3. Shekhar, S., Vatsavai, R., Chawla, S., Burke, T.E.: Spatial Pictogram Enhanced Conceptual Data Models and Their Translation to Logical Data Models, Integrated Spatial Databases, Digital Maps, and GIS, Lecture Notes in Computer Science, vol. 1737 (1999)
4. Shekhar, S., Chawla, S.: Spatial Databases: A Tour, Prentice Hall, Upper Saddle River, NJ, USA (2003) ISBN-130: 978-0130174802
5. Hoel, E.G., Heng, W.L., Honeycutt, D.: High Performance Multimodal Networks. In: Proceedings of International Symposium on Spatial and Temporal Databases SSTD (2005)
6. Shekhar, S., Liu, D.R.: CCAM: A Connectivity-Clustered Access Method for Networks and Network Computations. IEEE Trans. Knowl. Data Eng. **9**(1), 102–119 (1997)
7. Kohler, E., Langtau, K., Skutella, M.: Time-Expanded Graphs for Flow-Dependent Transit Times. In: Proceedings of 10th Annual European Symposium on Algorithms (2002)
8. George, B., Shekhar, S.: Time-aggregated Graphs for Modeling Spatio-temporal Networks – An Extended Abstract. In: Proceedings of Workshops at International Conference on Conceptual Modeling(ER), November (2006)
9. Winter, S.: Modeling Costs of Turns in Route Planning. GeoInformatica **6**(4), 363–380 (2002)
10. Anez, J., de la Barra, T., Perez, B.: Dual Graph Representation of Transport Networks. Transp. Res. **30**(3), 209–216 (1996)
11. Caldwell, T.: On Finding minimum Routes in a Network with Turn Penalties. Commun. ACM **4**(2), 107–108 (1961)
12. TomTom, http://www.tomtom.com/ (2007)
13. Garmin, http://www.garmin.com/garmin/cms/site/us (2006)
14. Google Maps, http://maps.google.com (2007)
15. Mapquest, http://www.mapquest.com (2007)

Road Network Data Model

LAURYNAS SPEIČYS, CHRISTIAN S. JENSEN
Department of Computer Science, Aalborg University, Aalborg, Denmark

Synonyms

Transportation network model; Spatial network model; Network data model; Graph; Link-node model; Linear reference model

Definition

A road network data model is a notation that enables the modeling of pertinent aspects of a road-network infrastructure. Using such a notation, a schema of a road-network infrastructure may be designed. This schema may in turn be populated by data, yielding an instance that captures aspects of a specific road network.

Depending of the use context, different aspects of a road-network infrastructure are of interest. A road network data model may consist of several interrelated sub-models, each of which targets the capture of different, specific aspects of a road network. Important examples of such sub-models include the geographical and graph representations and linear referencing.

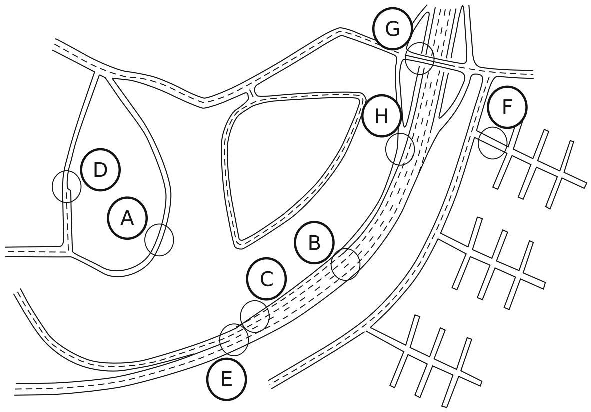
Geographical representations capture the embedding of a road network into geographical space. Specifically, a road is typically represented by a collection of polylines, where each polyline captures the centerline of part of a road.

Graph representations, also termed link-node representations, typically aim to capture the topology, or connectivity, of a road network in a compact and computationally efficient format. Graph representations are based on the concepts of undirected and directed mathematical graphs. A node, or vertex, typically models a location with a significant change of traffic properties. Such locations include road intersections. A link, or edge, models the part of the road network that enables travel between two nodes. In a directed-graph model, a directed edge captures that travel between the two nodes involved is allowed in the direction given by the edge. Edge weights capture travel distances or times. A binary so-called co-edge relation models the ability of vehicles to make u-turns in-between intersections. A binary so-called change-edge relation models the ability to make a lane change. To model roads with multiple lanes, multi-graphs that allow multiple edges between a pair of nodes are used.

With linear referencing, a road network is modeled as a collection of one-dimensional linear features that intersect at connections (locations where there is an exchange of traffic). With this model, any location in a road network is given as the identity of a linear feature and a distance from the start of the feature. Such locations may be used for the capture of content, including speed limits and accidents. The kilometer-post representation, an example of a known-marker representation, is often used for the specification of locations: a location is expressed in terms of the identifier of a road, a kilometer post (an example of a distance marker) on the road, and an offset from the distance marker.

With this model, it is also possible to associate a polyline with each linear feature that captures the embedding of the part of the road network modeled by the feature into geographical space.

Sub-models may be integrated by means of procedures that map instances of one sub-model to instances of another sub-model.



Road Network Data Model, Figure 1 Example road network depicting: (A) single-lane, one-way road, (B) multiple lanes in two directions, (C) emergence of a new lane, (D) abrupt change of a bi-directional road into a one-directional road, (E) split of a road into two, (F) restriction-free traffic in a residential area, (G) restricted u-turn, and (H) restricted lane change

Historical Background

Within computer science, the classical approach to the modeling of road networks is to use some variant of either a non-directed or a directed graph.

Both enable the capture of fundamental aspects of a road network, and are mathematically simple and compact formats that are conducive to efficient computation. The prototypical problem addressed in this setting is that of finding shortest paths between locations. In recent work, graphs with different geographical embeddings have been used for the modeling of road networks with the purpose of processing a variety of spatial queries, including variants of range and nearest neighbor queries (e. g., [1,8,11,13]).

So-called navigable data models have been created to support vehicle navigation. Only a few such models capture aspects such as lanes and the connectivities among lanes. Planar [4] and non-planar [3] models have been suggested. The non-planar model, which offers support for data maintenance, captures the geo-location and topology of lanes. The topological information consists of lane connectivities and turn restrictions along lanes and at impedance points. This information can then be used for constructing a graph on which a search can be performed.

In the domain of transportation where, typically, public authorities are concerned with the management (e. g., maintenance) of road-network infrastructures, focus has been on the development of data models that enable the convenient capture and maintenance of road-network related data [2,5,6]. The main objective is to provide means of managing content relevant to administrative tasks.

In this domain, linear referencing [14] has been used quite widely for the capture of content. As an indication of the importance of this domain, Oracle Spatial [10] offers support for linear referencing. In addition, generic schemas, in the form of ER diagrams, have been developed and recommended for the capture of different aspects of entire transportation infrastructures and related content [2,7,9,12,15].

Scientific Fundamentals

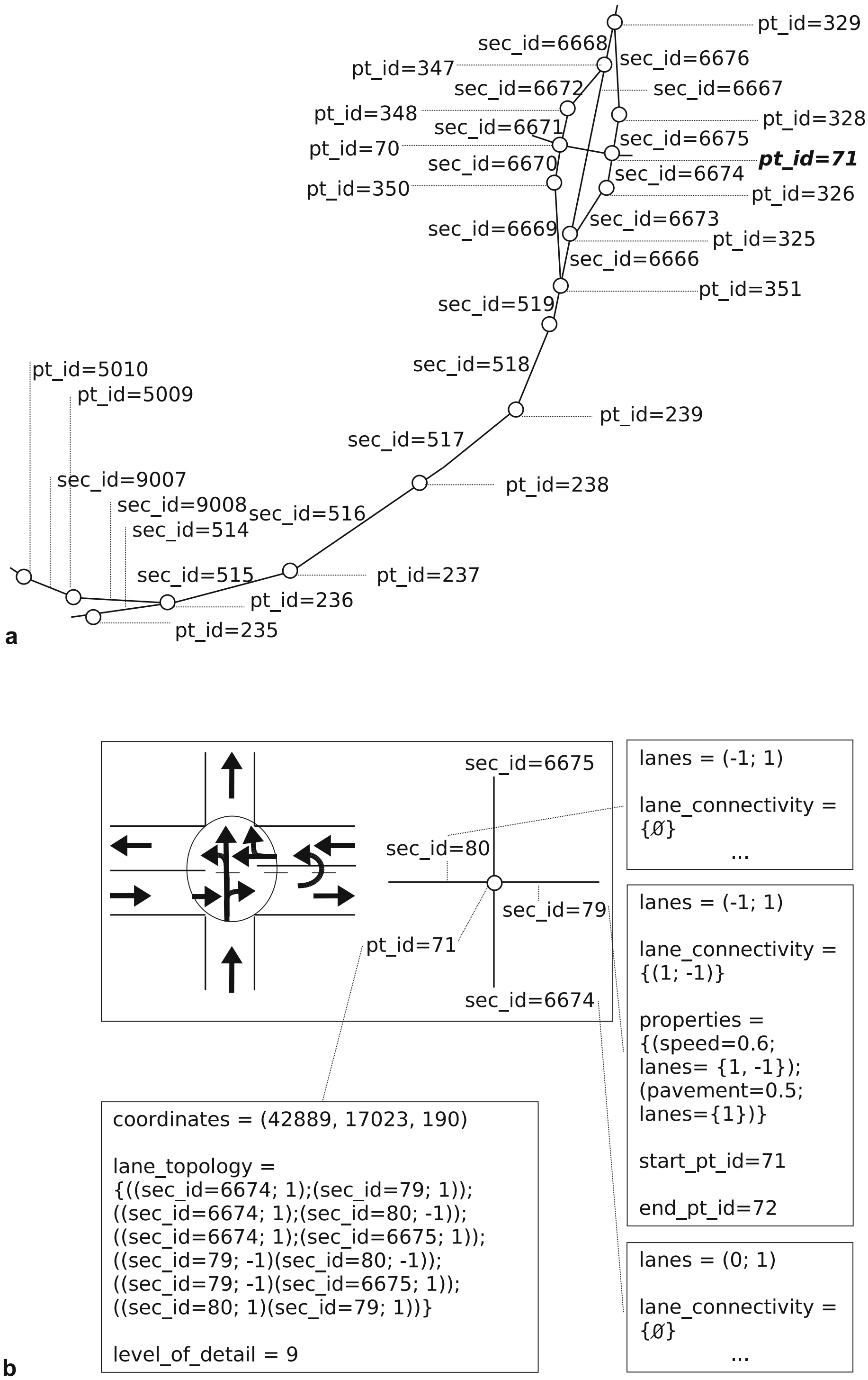
Example Road Network

The road network illustrated in Fig. 1 is used for exemplification. The figure depicts a sample road network that embodies a few of the aspects that a road network data model must be capable of capturing. In particular, this network illustrates some of the possible configurations of roads and lanes that are possible in real road networks. Specifically, the network includes bi-directional roads with several lanes in each direction, the appearance and disappearance of a lane due to local access to a highway, and the abrupt disappearance of lanes, i. e., a bi-directional road turning into a single-directional road without reaching an intersection and a road splitting into two.

The figure also illustrates different lane change restrictions. The residential areas to the right have no restrictions on lane changes and u-turns. However, u-turns are not allowed on the bridge that crosses the highway (top right); and on the highway, lane change is prohibited from the access lane for local traffic.

Geographical Road Network Data Models

Geographical data models are used for geo-referencing of the road infrastructure and thereby the road-related content. Specifically, three-dimensional points are the key



Road Network Data Model, Figure 2 Geographical model instance

building blocks of these models. Such points are typically used for the modeling of the center lines of roads. Pairs of such points thus define line segments that model sections of roads. A geographical representation of part of the example road network is shown in Fig. 2a. In the figure, the points are simply associated with identifiers—the coordinates are omitted.

It is often relevant to be able to represent the geography of a road network at different levels of detail. To enable that, an integer value capturing a level of detail may be associated with each point. The higher the number, the higher the level of detail. Thus, to obtain the representation of a road network at a level of detail corresponding to a certain number, all points with a level of detail number that does not exceed that of the chosen level of detail are considered. As a consequence, the entire sequence of points representing a road network is used when the highest level of detail is chosen.

As mentioned, pairs of consecutive coordinate points model road sections. (Note that consecutive points are allowed to have different levels of detail.) Thus, a road network is partitioned into small sections. In Fig. 2, these are also associated with identifiers. It is possible to associated traffic regulations with each section. For example, it is possible to capture the numbers of lanes in each direction for the section. The allowed movements from one lane to a neighboring lane can also be captured: movement between a pair of lanes may be prohibited/impossible or allowed from one lane to the other only or in both directions.

In an example approach, the lanes in a section are numbered by positive and negative integers, so that the lanes in the one direction are numbered 1, 2, etc. starting with the lane closest to the middle of the road. Similarly, the lanes in the other direction are numbered -1, -2, etc. The number 0 is used to indicate the absence of a lane in one direction. For example, Fig. 2b denotes section 6674 having 1 and 0 lanes in the one and in the other direction, respectively.

For each section, it is also possible to capture properties that affect the movement along the section. Movement-affecting properties are captured if their effect on movement can be quantified. Such properties include speed limits and congestion information. These and a spectrum of other properties can be quantified as conditions that hinder or facilitate movement with respect to some nominal movement condition. The effects of such properties can be normalized over the network. Figure 2b, here, includes a "pavement" property that affects the movement on lane 1 of section 79 by a factor of 0.5 of the nominal movement condition.

Coordinate points either connect two or more sections, or they mark a dead end of a road. For each coordinate point, it is possible to describe how traffic flows between the delimited sections. The possibility to move from a specific lane belonging to one section to a specific lane belonging to the other section is captured by an ordered pair where the first element identifies the former lane and the second element identifies the latter lane. Figure 2b illustrates how turn restrictions between lanes at an intersection are modeled. The figure shows traffic flowing in all directions excluding two u-turns (initiated from section 79, lane -1, and section 80, lane 1) and the left turn from section 80, lane 1 to section 6675, lane 1. Another situation where the description of flow is useful occurs at dedicated spots for u-turn, e. g., on highways with a separating line.

A less obvious use of turn restrictions occurs at points where a coordinate point is placed due to a configuration change on a road. Here lane directions, in most cases, do not change; however, the count of lanes in either of the directions may increase or decrease and the accessibility between lanes may change.

When a coordinate point is present only in order to capture the geographical embedding of a road network at a higher level of detail, the associated flow information is redundant in the sense that removing the coordinate point and its flow information does not result in a loss of flow information.

While the turn restrictions at a coordinate point, e. g., an intersection, can have many configurations, they must satisfy a few requirements. First, for each lane that allows traffic movement into the coordinate point, there must be at least one lane that accepts the incoming traffic; the inverse must be true as well; second, movement coming from a lane that starts at a coordinate point is prohibited, and movement to a lane that ends at a coordinate point is prohibited over the coordinate point.

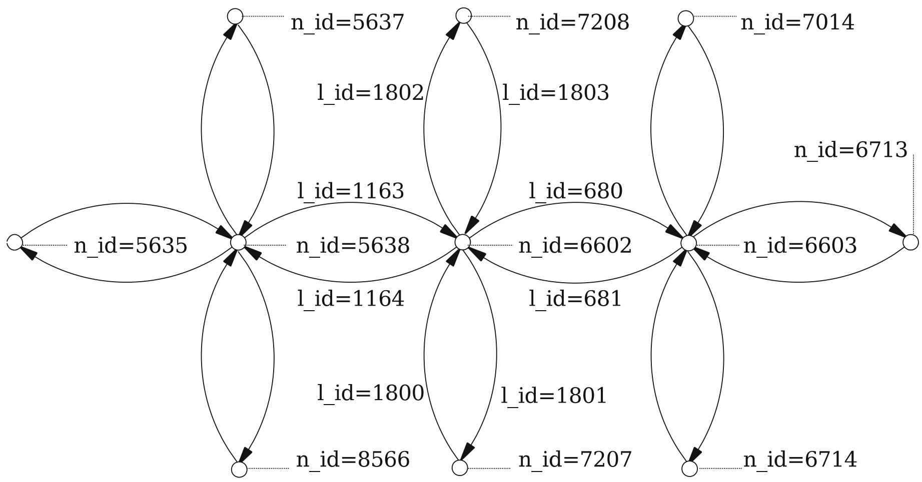
Link-Node Data Models

Link-node, or graph, data models abstract away geographical detail and aim to capture only the topology of a road network. The representations of road networks obtained by using these kinds of models are compact often very well suited as a foundation for query processing.

A typical link-node model captures a road network as a a collection of nodes connected by directed links, i. e., as a directed graph—Fig. 3 illustrates an example instance. (While undirected graphs may be used, they are not considered here.)

A road network may be modeled by several link-node instances that capture the road network at different levels of detail. For example, a certain region may be modeled by two link-node instances: a very detailed one used for detailed route planning and a less detailed one used for, e. g., higher-level route planning.

Thus, a node may be used in one or several instances. Conceptually, the use of a node is identified by a pair of attributes: a unique network identifier and a unique node identifier within the scope of the network. Similarly, a link may be used in several instances. Thus, usages of links have attributes analogous to those of nodes, i. e., the pair of a network identifier and a link identifier within the instance. Moreover, each link has a start node and an end node that belong to the same instance. Further, each



Road Network Data Model, Figure 3 A link-node model instance

link has a length that is normally constrained to be non-negative.

The link-node instance in Fig. 3 captures part of the road network depicted in Fig. 4 in a manner appropriate for high-level route planning. The links represent the routes, not individual roads, e. g., links 1163 and 1164 represent the forward and backward routes between the first and second intersections, respectively. For this reason, the complex intersections, i. e., the two over-passes and the rotary are each reduced to a single node, i. e., to the nodes 5638, 6602, and 6603.

In the example, the length for each link is approximately equal to the length of the corresponding route (in meters). In general, the length values may be given more complex semantics, e. g., the minimum travel time that is needed to traverse the corresponding route. A directed link in the example indicates that one node can be reached from another node. Thus, a route that corresponds to a bi-directional road is represented by a pair of oppositely directed links.

To capture a road network in a format that is appropriate for low-level navigation, a simple directed graph is not adequate. Instead, a directed multi-graph, i. e., a directed graph that can have more than one link (in the same direction) between a pair of nodes, is used.

A route followed by a vehicle through a road network may be captured using a sequence of full edges. However, it is at times allowed for a vehicle to make a u-turn, or to change lanes in-between nodes. To capture such behavior more accurately, one can introduce so-called co-edge and change-edge functions that model u-turns and changes of lanes.

A co-edge function captures pairs of edges that represent pairs of lanes for which it is allowed to make a u-turn from the first lane of the pair to the second lane. Intuitively, a moving object may jump from an edge to its co-edge without having reached a vertex. A u-turn can be made only to the opposing traffic lane that is closest to the current traffic direction. A moving object may overtake a car by temporarily entering an oppositely directed lane if that lane is a co-edge of the object's current lane.

Similarly, the change-edge function captures pairs of edges for which it is possible to change from the argument lane to the result lane. Lane changes are restricted to lanes in the same traffic direction. The intuition is the same as for u-turns—that of a moving object being able to jump from one lane to another without visiting a vertex.

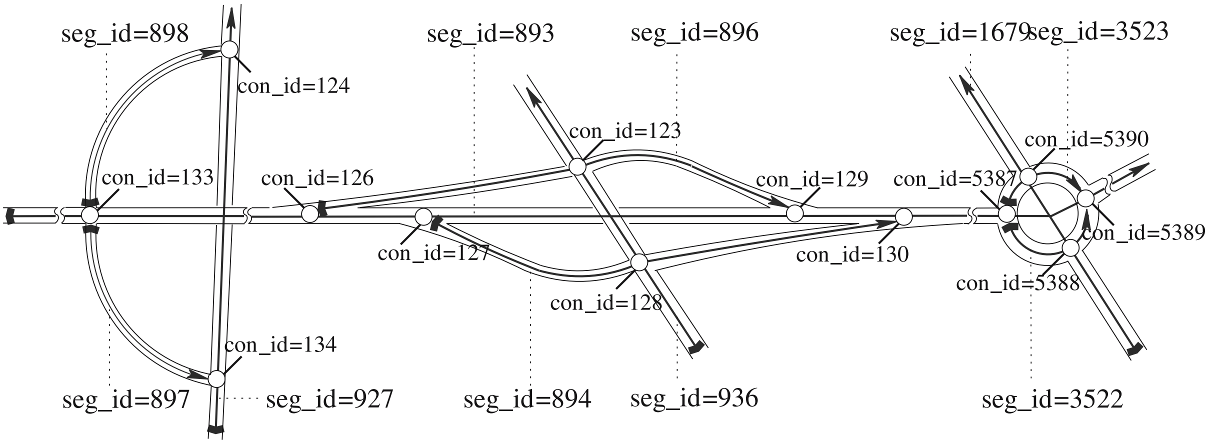
Linear Referencing Data Model

A linear referencing data model of a road network is a collection of one-dimensional so-called linear features, that intersect at connections. This representation is illustrated graphically in Fig. 4.

Each linear feature has a unique identifier and a length, and any position on the feature can be expressed as a distance from the start of the feature. The length of a linear feature is equal to the length of the corresponding part of the road network. The embedding of a linear feature into space can be, and typically is, described by a polyline, in a manner similar to what was described for the geographical data models.

Connections model intersections and also have unique identifiers. Several linear features are involved in a connection. For each such feature, the connection occurs at some distance from the start of the feature. It should be noted that the positioning of content, e. g., connections and accident reports, along a feature is independent of the polyline chosen for capturing the geographical location of the feature.

Figure 4 depicts an example of linear referencing-based model of part of a real road network [5]. In the example,



Road Network Data Model, Figure 4 Linear feature road network instance

the road network is modeled at a high level of detail. Each linear feature generally is as long as possible while preserving the network topology. For example, segment 893 corresponds to the "long" main road and is 78,326 meters long. In order to preserve the topology of the network, segment 3522, which is only 62 meters long, is assigned to the bottom semi-circumference of the rotary to the right, which connects two disjoint sections of the main road. As can be seen, connections are placed at road intersections.

When modeling a road network using linear referencing, the linear features should partition the road network. Long features are preferable because they lead to a more compact segment representation and, more importantly, a more compact and thus update-friendly representation of the associated content.

Key Applications

Typical applications of road network data models relate to different aspects of transportation.

Road Planning

Link-node data models are used for road planning, which refers to the (re)designing of road networks while taking, e. g., traffic flows into consideration.

Road Management

Kilometer-post based models (the most commonly used type of known-marker based models) are used for road administration. This type of model is useful for collecting and utilizing data in the field.

With this type of model, a location expressed in terms of a road, a distance marker on the road (i. e., a kilometer post), and an offset from the distance marker can be used for uniquely positioning of content. Primitive technological means, such as a simple measuring device and a map and a ruler, then suffice for identifying a position on a road.

Route Planning Services

Link-node type models are used for tasks such as route planning. The task refers to the retrieval of traversable routes that satisfy certain criteria from a road network. Directed graphs that capture traffic regulations are appropriate for this task.

In-Vehicle Navigation

One class of in-vehicle services concerns navigation. These services determine the most suitable route to a destination and provide instructions for real time navigation. For high-quality services, instructions take into account traffic regulations at the granularity of lanes. Another class of in-vehicle services relate to safety. Here, services warn drivers about possible collisions and assist the drivers in various ways. In order to provide this functionality, the services may rely on models that capture the underlying road network at lane granularity.

Mobile E-Services

Internet-worked mobile devices such as mobile phones, personal digital assistants, and navigation devices enable a range of new personal information services, many of which will exploit information about the user's geo-location for providing the desired functionality.

Example services include finder applications that allow the users to locate friends or family, businesses, or landmarks. Services may also deliver maps, directions, or traffic reports. Services may involve the identification of a service user's nearest neighbors of a certain kind. An example service may identify the emergency room that is within the closest driving distance; another may reserve the taxi that is nearest to the service user.

Such services concern objects moving in and located in road networks, as well as content reachable via road networks. Thus, road network models are needed for the underlying computations.

Future Directions

The detailed modeling of road networks is becoming increasingly relevant. Specifically, advances in positioning technologies are slated to enable the positioning of vehicles within lanes.

For example, the Galileo positioning system will offer better positioning than does GPS with respect to several aspects, including the accuracy, penetration, and time to fix. For example, the best-case accuracy (without the use of ground stations) of Galileo is 45 cm as opposed to 2 m for GPS. Next generation GPS will also offer better positioning, and Galileo and GPS are expected to be interoperable. As another example, infrastructures that rely on in-road and in-vehicle sensors for accurate positioning at lane resolution are being conceived in the telematics community.

Exploitation of the capability of positioning at lane resolution will enable increases in the qualities of existing services, e. g., navigation services, but will also enable entirely new services, e. g., collision warning and assisted driving services.

Road network data models that capture road-network infrastructures at the lane level are essential in many applications that may exploit lane-level positioning. Such models serves as a basis for data retrieval, i. e., for query processing, which is the topic of much recent and on-going work. For example, with the objective of supporting different location-based services, much attention is being dedicated to the efficient support for a variety of proximity queries—including conventional, skyline, and trajectory-based queries, for static as well as moving objects. This line of research may lead to both future modifications of the model, to new insights into query processing, and to new and exciting applications.

Cross References

► ArcGIS: General Purpose GIS Software System
► Contraflow in Transportation Network
► Data Compression for Network GIS
► Emergency Evacuations, Transportation Networks
► Location-Based Services: Practices and Products
► Nearest Neighbor Queries in Network Databases
► Network GIS Performance
► Oracle Spatial, Geometries
► Road Maps, Digital
► Routing Vehicles, Algorithms
► Routing Vehicles, Algorithms
► Trip Planning Queries in Road Network Databases
► Voronoi Diagrams for Query Processing

Recommended Reading

1. De Almeida, V.T., Güting, R.H.: Using Dijkstra's Algorithm to Incrementally Find the k-Nearest Neighbors in Spatial Network Databases, In: Proc. ACM Symposyum on Apagesied Computing, pp. 58–62 (2006)
2. Dueker, K., Butler, J.A.: GIS-T Enterprise Data Model with Suggested Implementation Choices, J. Urban Reg. Inform. Syst. Assoc. **10**(1), 12–36 (1998)
3. Fohl, P., Curtin, K.M., Goodchild, M.F., Church, R.L.: A Non-Planar, Lane-Based Navigable Data Model for ITS, In: Proc. International Symposium on Spatial Data Handling, pp. 17–29 (1996)
4. Gottsegen, J., Goodchild, M., Church, R.: A Conceptual Navigable Database Model for Intelligent Vehicle Highway Systems, In: Proc. GIS/LIS, pp. 371–380 (1994)
5. Hage, C., Jensen, C.S., Pedersen, T.B., Speicys, L., Timko, I.: Integrated Data Management for Mobile Services in the Real World, In: Proc. International Conference on Very Large Data Bases, pp. 1019–1030 (2003)
6. Jensen C.S.: Database Aspects of Location-Based Services, In: Schiller, J., Voisard, A.: Location-Based Services, Morgan Kaufmann Publishers, pp. 115–148 (2004)
7. Koncz, N., Adams, T.M.: A Data Model for Multi-Dimensional Transportation Location Referencing Systems, J. Urban Reg. Inform. Syst. Assoc. **14**(2), 27–41 (2002)
8. Kolahdouzan, M., Shahabi, C.: Voronoi-Based K Nearest Neighbor Search for Spatial Network Databases, In: Proc. International Conference on Very Large Data Bases, pp. 840–851 (2004)
9. NCHRP: A Generic Data Model for Linear Referencing Systems, Transportation Research Board, Washington, DC (1997)
10. Murray, C.: Oracle Spatial User Guide and Reference, Release 9.2., Oracle Corporation (2002)
11. Papadias, D., Zhang, J., Mamoulis, N., Tao, Y.: Query Processing in Spatial Network Databases, In: Proc. International Conference on Very Large Data Bases, pp. 802–813 (2003)
12. Okunieff, P., Siegel, D., Miao, Q.: Location Referencing Methods for Intelligent Transportation Systems (ITS) User Services: Recommended Apagesoach, In: Proc. GIS-T Conference, pp. 57–75 (1995)
13. Sankaranarayanan, J., Alborzi, H., Samet, H.: Efficient Query Processing on Spatial Networks, In: Proc. ACM International Workshop on Geographical Information Systems, pp. 200–209 (2005)
14. Scarponcini, P.: Generalized Model for Linear Referencing in Transportation, GeoInformatica **6**(1), pp. 35–55 (2002)
15. Zeiler, M.: Modeling Our World, ESRI Press (1999)

Road Networks

► Contraflow in Transportation Network
► Spatio-temporal Queries on Road Networks, Coding Based Methods

Roadway Network Model

► Emergency Evacuations, Transportation Networks

Root-Mean-Square Error

► Photogrammetric Products

Rough Approximation

► Approximation

Rough Set Theory

► Approximation

Route Activity

► Crime Mapping and Analysis

Routing

► Data Collection, Reliable Real-Time

Routing Vehicles, Algorithms

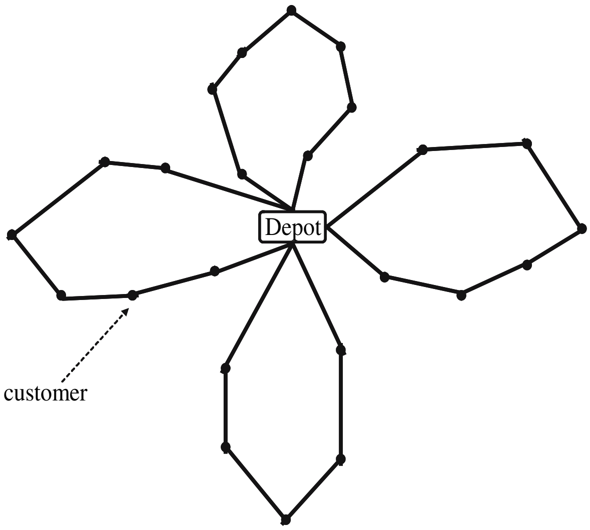
CHRISTOS D. TARANTILIS
Department of Management Science & Technology,
Athens University of Economics & Business,
Athens, Greece

Synonyms

Distribution logistics; Fleet management; Vehicle routing problem; Simulated annealing; Threshold accepting method; Record-to-record travel method

Definition

The Vehicle Routing Problem (VRP) [1] embraces a class of complex combinatorial optimization problems that target the derivation of minimum total cost routes for a number of resources (vehicles) located at a central point (depot) in order to service efficiently a number of demand points (customers). The standard version of VRP (known as basic VRP) is defined on a graph $G = (V,A)$, where



Routing Vehicles, Algorithms, Figure 1 A typical VRP solution

$V = \{u_0, u_1, \ldots, u_n\}$ is the vertex set and $A = \{(u_i, u_j): u_i, u_j \in V, i \neq j\}$ is the arc set of G. Vertex u_0 represents a depot (warehouse or distribution centre) that hosts a homogeneous fleet of m vehicles with capacity Q. The remaining vertices correspond to demand points (or equivalently, customers). Each customer u_i has a non-negative demand q_i. The vector of all customer demands is denoted by $q(V)$. Furthermore, a non-negative cost matrix $C = (c_{ij})$ is defined on A; usually, the cost c_{ij} models the travel time between customers u_i and u_j. If $c_{ij} = c_{ji}$, the problem is symmetric, and it is common to replace A with the edge set $E = \{(u_i, u_j): u_i, u_j \in V, i \not< j\}$. The solution to the basic VRP is a set of routes that satisfy the following constraints: a) each route starts and ends at the central depot; b) each customer is visited exactly once; c) every customer's demand is satisfied; d) the total travel time of the set of routes is minimized.

The aim of this chapter is to focus on the annealing-based solution approaches for solving two of the most studied VRP types: the Capacitated VRP (CVRP) and the Distance Constrained VRP (DCVRP) [2]. The additional constraints imposed to model the routing scenarios of the aforementioned VRP types are:

- the total demand of the customers covered by a route cannot exceed the capacity of a vehicle Q (for both CVRP and DCVRP);
- the total travel time of any vehicle route cannot exceed a pre set upper bound (only for DCVRP).

The objective (for both CVRP and DCVRP) is to minimize the sum of travel time. It is important to note that the number of vehicles is either pre-determined or is treated as a decision variable.

R

Historical Background

In the late 70s, several algorithms were developed for solving VRPs for very small numbers of variables and constraints. Later, since exact algorithms were not capable of consistently solving instances with more than 50 customers, heuristics were mainly employed for real-life medium and large-scale vehicle routing problems [3].

Heuristics can be divided into two classes: *classical heuristics* that perform relatively limited exploration of the search space to produce good solutions fast, and *metaheuristics* that are general-purpose mechanisms guiding intelligently the search process, combining neighborhood search rules, memory structures and recombination of solutions [4].

Classical heuristics for both the CVRP and DCVRP can be broadly classified into three categories [3]:

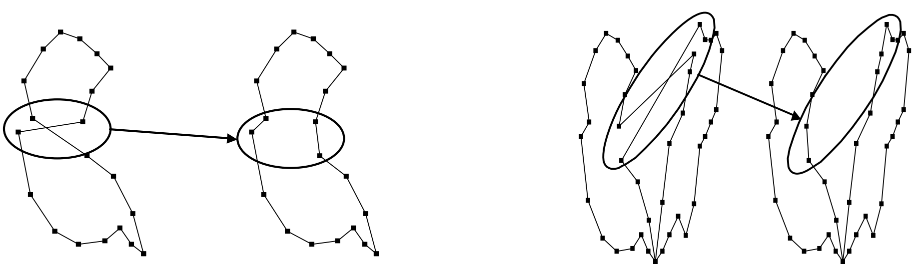
a) *Construction heuristics* build (guided by some cost minimization criterion) a solution, selecting a solution component (vertex or arc) step by step until the partial solution is completed. The solution can be constructed sequentially (i. e. producing one route at a time) or parallel (i. e. producing several routes simultaneously). The myopic approach of adding the best solution component according to the least cost increase is called *greedy* approach.

b) *Two-phase heuristics* produce a feasible solution in the first phase and then optimize the sequence of customers on each route in the second phase.

c) *Local search classical heuristics* used to improve a VRP-solution by introducing changes in the current solution. Local search is an iterative search procedure that, starting from an initial feasible solution, progressively improves it by applying a series of local modifications called *moves*. At each iteration of local search, the set of *moves* that can be applied to the current solution s, define a set of neighboring solutions denoted $N(s)$. More specifically, $N(s)$ is a subset of search space (i. e. space of all possible solutions than can be visited during the search) which consists of solutions generated by applying a single transformation to the current solution s. According to local search rationale, at each iteration, the search moves to an improving neighbour-feasible solution until it will get trapped in a local optimum, which usually represents a low quality solution.

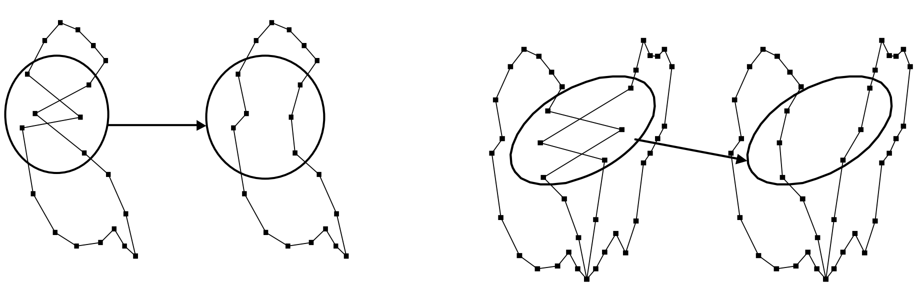
Three of the most applicable moves are the *2-Opt*, 1–1, and *1–0 Exchange* moves [3].

A 2-Opt *move* operates as follows: Suppose a route $R = \{u_0, u_1, \ldots, u_n, u_0\}$ of a solution s, and let $Z = \{(u_i, u_{i+1}); (u_j, u_{j+1})\}$ be a set of two edges in R that form criss-cross. A 2-Opt *move* eliminates then the criss-cross and reverses a section of the R by replacing the edges of Z by edges of $W = \{(u_i, u_j); (u_{i+1}, u_{j+1})\}$ to reconstruct the route R. In the multiple routes, edges (u_i, u_{i+1}) and (u_j, u_{j+1}) belong to different routes but they form a criss-cross again. A 2-Opt *move* is applied exactly in the same way as in the case of single route. This is demonstrated in Fig. 2 [2].



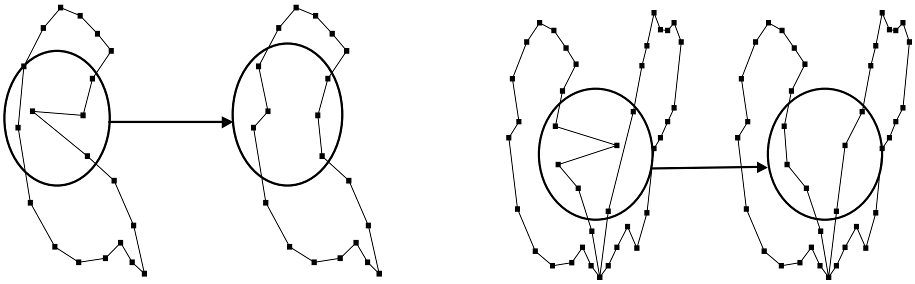
2-Opt move for single route 2-Opt move for multiple routes

Routing Vehicles, Algorithms, Figure 2 The 2-Opt move



1-1 Exchange move for single route 1-1 Exchange move for multiple routes

Routing Vehicles, Algorithms, Figure 3 The 1–1 Exchange move



Routing Vehicles, Algorithms, Figure 4 The 1–0 Exchange move

In 1–1 Exchange *move*, two vertices from either the same or different routes are swapped, while in 1–0 Exchange *move*, a vertex is transferred from its position in one route to another position in either the same or a different route. These moves are shown in Figs. 3 and 4 respectively [2].

As a major departure for the classical heuristics, metaheuristics are non problem specific approximate algorithms either stochastic or deterministic which combine, guide and subordinate classical heuristic methods in higher level frameworks, using different types of memory structures and learning mechanisms, as well as analogies with optimization methods found in nature. Metaheuristics typically produce much higher quality solutions than those obtained with classical heuristic approaches [5].

Annealing-based algorithms belong to the first metaheuristics extending classical local search by employing explicit strategies to escape from local optima, allowing moves to inferior solutions. The term "annealing" is due to the fact that conceptually the algorithms of this category inspired by a physical process, known as annealing, where a material is heated into a liquid state then cooled back into a recrystallized solid state [6]. The main advantages of annealing-based algorithms are their simple structure, general applicability and computational effectiveness on different combinatorial optimization problems.

In the following section of this chapter, the implementations of the most effective annealing-based metaheuristics to the solution of the CVRP and DCVRP are reported.

Scientific Fundamentals

A description of the basic principles of the annealing-based metaheuristic approaches is given first, followed by a description of the implementations to the CVRP and DCVRP.

Simulated Annealing – Basic Principles

Simulated Annealing (SA) [6] finds its inspiration by physical annealing process studied in statistical mechanics. SA conducts local search while offering the possibility of accepting, in a controlled manner, worse solutions. This feature allows SA to escape from a low quality local optimum.

More precisely, at each iteration t of SA, a neighbour $s' \in N(s)$ of the current solution s is generated stochastically and a decision is then made to decide whether s' will replace s. If s' is better than s i. e. $\Delta = c(s') - c(s) \leq 0$ (for a minimization problem), the search moves from s to s', otherwise, the search is moved to s' with the probability $e^{(-\Delta)/\theta t}$. This probability depends on a) the degree of the degradation, and b) a control parameter θ called temperature (higher temperatures lead to higher accepting probabilities and vice versa). The temperature is controlled by a cooling schedule specifying how the temperature should be progressively reduced. Typically, SA stops when a fixed number of non-improving iterations is realized with the temperature or when a pre-specified number of iterations is reached. The SA algorithm is summarized in the Fig. 5.

R

Simulated Annealing Algorithms for the CVRP and DCVRP

Osman [7] implemented a SA algorithm for solving the CVRP and DCVRP, using a systematic way to explore the neighbourhoods of the current solution. Route pairings were first constructed in the following order: $(R_{\pi(1)}, R_{\pi(2)}), \ldots, (R_{\pi(1)}, R_{\pi(m)}), (R_{\pi(2)}, R_{\pi(3)}), \ldots, (R_{\pi(m-1)}, R_{\pi(m)})$ where π was a permutation of customers and m was the number of vehicles used. Then, a λ-*interchange* neighbourhood structure was employed, in which exchanges of customers for each pairing took place until an improvement was identified. A sophisticated cooling schedule was also applied: the temperature θ_t at iteration t decreased continuously as long as the current solution was modified, while it was either halved or replaced by the temperature at which the incumbent was identified whenever the local search in the inner loop was completed without accepting any λ-*interchange* move.

Simulated Annealing

Initialization: Step 1. Produce an initial feasible solution s

Step 2. Select an initial temperature $\theta_1 > 0$.

Outer Loop: While Outer Loop criterion NOT satisfied do

Inner Loop: While Inner Loop criterion NOT satisfied do

i) Generate solution $s' \in N(s)$);

ii) If $\Delta = c(s') - c(s) \leq 0$, then // *Acceptance*

{set $s = s'$; check if $c(s) < c(s_{best})$ then $s_{best} = s$;}

iii) If $\Delta = c(s') - c(s) > 0$, then $s = s'$ with probability $\exp(-\Delta / \theta_t)$.

Repeat Inner Loop:

If s' replaces s at least once in inner loop,

then decrease temperature θ_t { $\theta_t = \theta_{t+1}$; $\theta_{t+1} = \alpha_1 * \theta_t$; }

Repeat Outer Loop:

Report best solution found;

Routing Vehicles, Algorithms, Figure 5 The Simulated Annealing algorithm

Threshold Accepting

Initialization: Step 1. Produce an initial feasible solution s

Step 2. Select an initial threshold $T_{ho} > 0$.

Outer Loop: While Outer Loop criterion NOT satisfied do

Inner Loop: While Inner Loop criterion NOT satisfied do

Generate solution $s' \in N(s)$;

If $c(s') - c(s) \leq T_h$, then // *Acceptance*

{set $s = s'$; check if $c(s) < c(s_{best})$ then $s_{best} = s$;}

Repeat Inner Loop.

If s' replaces s, i.e. if $c(s') - c(s) \leq T_h$ is satisfied at least once in inner loop, then reduce threshold

{ $T_{old} = T_{new}$; $T_{new} = \alpha_1 * T_{old}$; } // Threshold reduction

Repeat Outer Loop.

Report best solution found;

Routing Vehicles, Algorithms, Figure 6 The Threshold Accepting algorithm

Threshold Accepting – Basic Principles

Threshold accepting (TA) [8] is a modification of the SA. More precisely, it leaves out the stochastic element in accepting worse solutions by introducing a deterministic threshold, denoted $T_h > 0$, and accept a worse solution if $c(s') - c(s) \leq T_h$ (the inequality represents the *move acceptance criterion*). During the optimization process the threshold level is gradually lowered like the temperature in SA. As long as the value of T_h is high, the local search performed is not goal oriented, thus achieving high diversification (i. e. elaborating different regions in the solution space) and low intensification (i. e. concentrating the search into a specific region of the search space) of the search process. However, as the search procedure evolves and T_h is reduced, the balance between diversification and intensification changes until the typical threshold accepting algorithm behaves nearly like a descending local search algorithm (i. e. accepts only cost-improving solutions). The TA algorithm is summarized in the Fig. 6.

Threshold Accepting Algorithms for CVRP and DCVRP

Tarantilis et al. [9,10] developed two TA-based algorithms for solving CVRP and DCVRP called Backtracking Adaptive Threshold Accepting (BATA) and List Based Threshold Accepting (LBTA) respectively. The basic innovation of BATA over the standard TA method was that T_h was not necessarily reduced in a monotonic fashion during the search process but also incorporated an occasional increase, called backtracking, of its value. The adopting of

BATA

Initialization: Step 1. Produce an initial feasible solution s

Step 2. Select an initial threshold $T_{ho} > 0$.

Outer Loop: While Outer Loop criterion NOT satisfied do

Inner Loop: While Inner Loop criterion NOT satisfied do

Generate solution $s' \in N(s)$;

If $c(s') - c(s) \leq T_h$, then // *Acceptance*

{set $s = s'$; check if $c(s) < c(s_{best})$ then $s_{best} = s$;}

Repeat Inner Loop.

If s' replaces s, i.e. if $c(s') - c(s) \leq T_h$ is satisfied at least once in inner loop, then reduce threshold

{ $T_{old} = T_{new}$; $T_{new} = \alpha_1 * T_{old}$; } // Threshold reduction

else

raise the threshold value (backtracking)

Repeat Outer Loop.

Report best solution found;

LBTA

Initialization: Step 1. Produce an initial feasible solution

Step 2. Compute the threshold values that represent the initial List by conducting local search

Loop: While Loop criterion NOT satisfied do

New acceptances based the maximum threshold value, $T_{\max}$, stored in the List at every iteration

(a) Generate s' from s. Compute $T_{new} = \dfrac{c(s') - c(s)}{c(s)}$

(b) If $T_{new} = \dfrac{c(s') - c(s)}{c(s)} \leq T_{\max}$ then {set $s = s'$; check if $c(s) < c(s_{best})$ then $s_{best} = s$;}. If (s has changed then insert T_{new} in List) {Insert: $T_{new} \rightarrow$ *List;* Pop: *List* $\rightarrow T_{max}$}

(c) Repeat **Loop**;

Report best solution found;

Routing Vehicles, Algorithms, Figure 7 The BATA and the LBTA algorithms

Record to Record Travel

Initialization: Step 1. Produce an initial feasible solution s

Step 2. Select a Deviation $D > c(s)$. Let $c(s_{best}) := c(s)$

Loop: While stop criterion NOT satisfied do

i) Generate solution $s' \in N(s)$;

ii) If $c(s') < c_{best} + D$ then // *Acceptance*

{set $s = s'$; check if $c(s) < c(s_{best})$ then $s_{best} = s$;}

Repeat Loop.

Report best solution found;

Routing Vehicles, Algorithms, Figure 8 The Record-to-Record Travel algorithm

this non-monotonic schedule, results in an oscillating strategy that achieved a dynamic balance between diversification and intensification of the search process. Regarding LBTA, its basic innovation over the standard TA method was the introduction of a List of threshold values, which were used in the *move acceptance criterion* expressed as $\frac{c(s')-c(s)}{c(s)} \leq T_{\max}$ where $T_{\max}$ was the maximum threshold value stored in the list, and helped the method decide whether s' will replace s.

More precisely, the List served as a memory of the variability of local function values, stored in the form of value changes from each old configuration. The introduction of the List of threshold values also helped the designer of a VRP-algorithm reduce the parameters involved in the threshold reduction strategy, since parameters (such as the initial value of threshold and the percentage of the threshold reduction) were determined automatically.

Record-to-Record Travel – Basic Principles

The Record-to-Record Travel algorithm [11] accepts the neighbor s' of the current solution if it is not much worse than the best solution found so far during the optimization process. The cost of the best solution is called *Record*. The *Deviation* is the only parameter of the algorithm, defined as $k\% \times Record$. The Record-to-Record Travel algorithm is summarized in the Fig. 8.

Record-to-Record Travel Algorithms for CVRP and DCVRP

Golden et al. [12] developed a Record-to-Record Travel (RtRT) algorithm for solving large-scale vehicle routing problems. The proposed methodology constructed an initial solution by applying the Clarke and Wright savings heuristic [3]. The 1–0 and 1–1 exchange feasible moves were then employed within the record-to-record travel procedure. To clean up routes produced, only cost-improving moves were then used. The individual routes were then re-sequenced and the operations of 1–0 exchange, 1–1 exchange and clean-up were repeated. Whenever the solution has not improved for a number of iterations, the best solution was perturbed by reinserting some of its vertices in different positions and repeating the operations of 1–0, 1–1 exchanges and clean-up. The Golden et al. RtRT algorithm is sketched in Fig. 9.

The RtRT metaheuristic developed by Li et al. (2004) [13] was a modification of the Golden et al. [12] algorithm. The algorithm was called VRtRT due to the variable-length neighbor list used within the RtRT procedure. This idea, which was inspired by the Granular Tabu Search algorithm [4], was to (a priori) delete from the examined graph long edges that were unlikely to be part of the optimal solution. Following this rationale, the VRtRT examined only a fixed number of neighbors for each vertex when employing the operations of 1–0 exchange, 1–1 exchange and 2-opt moves. These neighbors determined by a proportion p of the 40 shortest edges incident to each vertex. The parameter p varied during the optimization process.

Computational Results

The metaheuristic algorithms described in this Section, as far as we know, constitute the most effective annealing-based algorithms (in terms of solution quality) for solving the CVRP and DCVRP, according to their performance on the following two well-known sets of benchmark instances:

- The fourteen benchmark instances generated by Christofides et al. [14]. Each instance contains between 50 and 199 nodes as well as the depot. The location of the nodes is defined by their Cartesian co-ordinates, and the travel cost from vertex u_i to u_j is assumed to be the respective Euclidean distance. Problems 1–5, 11 and 12 are CVRPs while the problems 6–10, 13 and 14 are DCVRPs. For the first ten problems, vertices are randomly located over a square, while for the remaining ones, vertices are distributed in clusters and the depot is not centered;
- The twenty large scale vehicle routing problems (LSVRPs) proposed by Golden et al. [12]. These instances contain between 200 and 483 customers while the first eight of them have route-length restrictions (i. e. DCVRPs).

The problem instances can be found at http://neo.lcc.uma.es/radi-aeb/WebVRP/.

Close examination of the results demonstrated in Tables 1– 4 shows that the concept of VRtRT introduced by Li et al. [13] is probably the most powerful idea put forward in the area of annealing-based metaheuristics for solving CVRPs and DCVRPs in recent years. However, LBTA [10] manages to produce competitive results with VRtRT, using just one parameter within its structure.

It is also noteworthy that the variable-length neighbor list helps VRtRT both speed up the search process and produce

<u>The RtRT algorithm of Golden *et al* </u>[12]
Step 1. Produce an initial solution by Clarke and Wright
Step 2. Apply feasible 1-0 exchange moves and record-to-record travel.
Step 3. Apply feasible 1-1 exchange moves on different routes and record-to-record travel.
Step 4. Apply feasible 1-0 exchange, 1-1 exchange and 2-opt moves (only cost-improving moves)
Step 5. Apply local reinitialization: Repeat R_1 times and resequence individual routes. Go to Step 2.
Step 6. Repeat R_2 times and pertub the current best solution. Go to Step 2.

Routing Vehicles, Algorithms, Figure 9 The RtRT algorithm of Golden et al. [12]

Routing Vehicles, Algorithms, Table 1 Computational performance of the best known annealing-based metaheuristics on the benchmark instances generated by Christofides et al. [14]

Pr.	Osman	BATA	LBTA	RtRT	VRtRT
	Value	Value	Value	Value	Value
1	528	524.61	524.61	–	524.61
2	838.62	839.56	838.18	–	836.18
3	829.18	830.34	830.21	–	827.39
4	1058	1037.17	1036.05	–	1045.36
5	1378	1318.49	1317.81	–	1303.47
6	555.43	555.43	555.43	–	–
7	909.68	909.68	909.68	–	–
8	866.75	868.58	867.41	–	–
9	1164.12	1174.60	1173.89	–	–
10	1417.85	1418.27	1421.01	–	–
11	1176	1042.11	1042.11	–	1042.11
12	826	819.56	819.56	–	819.56
13	1545.98	1547.74	1547.28	–	–
14	890	866.37	866.37	–	–

Routing Vehicles, Algorithms, Table 2 Average computing comparison (in minutes) of four annealing-based meta- heuristics on the benchmark instances generated by Christofides et al. [14]

	Osman SAVAX 8600	BATAPentium II 400 MHz	LBTAPentium II 400 MHz	VRtRT Athlon 1 GHz
	Time	Time	Time	Time
Average Computing time	151.4	6.5	6.8	0.41

Routing Vehicles, Algorithms, Table 3 Performance of four annealing-based meta-heuristics on large scale problem instances generated by Golden et al. [12]

Pr.	Osman	BATA	LBTA	RtRT	VRtRT
	Value	Value	Value	Value	Value
1	–	5683.63	5680.16	5834.60	5666.42
2	–	8528.80	8512.64	9002.26	8469.32
3	–	11199.72	11190.38	11879.95	11145.80
4	–	13661.16	13706.78	14639.32	13758.08
5	–	6466.68	6460.98	6702.73	6478.09
6	–	8429.28	8427.72	9016.93	8539.61
7	–	10297.27	10274.19	11213.31	10289.72
8	–	11953.93	11968.93	12514.20	11920.52
9	–	596.92	595.35	587.09	588.25
10	–	765.03	764.88	749.15	749.49
11	–	945.20	945.09	934.33	925.91
12	–	1143.39	1143.74	1137.18	1128.03
13	–	872.66	871.97	881.04	865.20
14	–	1102.40	1102.66	1103.69	1097.78
15	–	1384.04	1385.59	1364.23	1361.41
16	–	1679.50	1677.25	1657.93	1635.58
17	–	718.16	717.40	720.44	711.74
18	–	1030.54	1032.07	1029.21	1010.32
19	–	1408.62	1406.47	1403.05	1382.59
20	–	1872.23	1872.87	1875.17	1850.92

Routing Vehicles, Algorithms, Table 4 Average computing comparison (in minutes) of four annealing-based meta- heuristics on large scale problem instances generated by Golden et al. [12]

	BATAPentium II 400 MHz Time	*LBTAPentium II 400 MHz Time*	*RtRT Pentium 100 MHz Time*	*VRtRT Athlon 1 GHz Time*
Average Computing time	18.4	17.8	37.2	1.13

much better results than the RtR algorithm (due to focus on promising moves).

Key Applications

Regarding the practical interest, in today's competitive business environment, VRP is not just a model of moving goods, but rather a critical model in effectively managing and operating the supply chain. In particular, VRP is involved in:

Transportation

The VRP model is one of the models that holds the supply chain together and companies that invest in research for developing effective algorithms will definitely improve their transportation management practices and will undoubtedly generate strategic benefits for themselves and their supply chain partners.

E-commerce

"People don't buy products, they buy delivered products," highlight supply chain experts. That's why it is essential to tie e-commerce to a distribution network. It is understandable that the growth of e-commerce and its distribution needs are inevitable. Customers are buying their products online because of convenience and speed of delivery. If e-commerce companies can't deliver products in a timely matter, customers either will buy from an online competitor or from the local retail store. In addition, as online sales continue to increase, more and more companies will be faced with the dilemma of partnering with a logistics company that will provide a cost-effective and seamless distribution solution. The companies that don't step up to the plate may find themselves as obsolete as the e-commerce companies that fail to create affordable and efficient distribution networks.

Reverse Logistics

In addition to the distribution process to the customers, re-usable packaging and goods to be recycled or remanufactured have to be transported in the reverse direction. On a strategic level, design decisions for the reverse logistics system have to be taken: Who performs which task and where. On a medium-term level the operator of the redistribution system and the relation between forward and reverse channel (thus distribution and redistribution system) have to be determined. If the decision is made that the foreword and reverse are to be run independently, for each of the channels a separate VRP has to be solved.

Green and Risk Logistics

The VRP methodology plays an important role in Green and Risk Logistics operations by finding routes for the transportation of hazardous materials (i. e. transportation of gas cylinders) [15] such that the population exposure risk or/and the of the environment pollution risk is mitigated.

Future Directions

The high quality solution produced by VRtRT, LBTA and BATA encourages for future work on designing annealing-based metaheuristics with simple structure, few parameters and intelligent strategies for solving real-time vehicle routing problems (RT-VRPs). These constitute a generic class of dynamic and mix of dynamic and stochastic VRPs aiming at quick reaction within a prescribed time schedule, in response to the incomplete or uncertain information revealed (customer requests, travel times, breakdowns, etc) as routes are executed.

Recommended Reading

1. Toth, P., Vigo, D.: The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia (2002)
2. Tarantilis, C.D., Kiranoudis, C.T.: Using a spatial decision support system for solving the vehicle routing problem. Inf. Manag. **39**(5), 359–375 (2002)
3. Laporte, G., Semet, F.: Classical heuristics for the capacitated VRP. In: Toth, P., Vigo, D. (eds.) The Vehicle Routing Problem, pp. 109–128. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia (2002)
4. Gendreau, M., Laporte, M.G., Potvin, J.-Y.: Metaheuristics for the capacitated VRP. In: Toth, P., Vigo, D. (eds.) The Vehicle Routing Problem, pp. 129–154. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia (2002)
5. Tarantilis, C.D., Gendreau, M., Spinellis, D.: The Impact of Metaheuristic Strategies in Transportation and Logistics Applications. IEEE Intelligent Syst. **20**(4), 16–18 (2005)

6. Kirkpatrick, S., Gelatt, C.D., Jr., Vecchi, M.P.: Optimization by simulated annealing. Science **220**, 671–680 (1983)
7. Osman, I.H.: Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Ann. Oper. Res. **41**, 421–451 (1993)
8. Dueck, G., Scheuer, T.: Threshold accepting. A general purpose optimization algorithm appearing superior to simulated annealing. J. Comput. Phys. **90**, 161–175 (1990)
9. Tarantilis, C.D., Kiranoudis, C.T., Vassiliadis, V.S.: A backtracking adaptive threshold accepting metaheuristic method for the Vehicle Routing Problem. Syst. Anal. Model. Simul. **42**(5), 631–644 (2002)
10. Tarantilis, C.D., Kiranoudis, C.T., Vassiliadis, V.S.: A list based threshold accepting algorithm for the capacitated vehicle routing problem. Int. J. Comput. Math. **79**(5), 537–553 (2002)
11. Dueck, G.: New optimization heuristics: the great deluge algorithm and the record to record travel. J. Comput. Phys. **104**, 86–92 (1993)
12. Golden, B.L., Wasil, E.A., Kelly, J.P., Chao, I.-M.: The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results. In: Crainic, T., Laporte, G. (eds.) Fleet management and logistics, pp. 33–56. Kluwer, Boston (1998)
13. Li, F., Golden, B.L., Wasil, E.A.: Very large-scale vehicle routing: New test problems, algorithms, and results. In: Computers & Operations Research, Forthcoming (2004)
14. Christofides, N., Mingozzi, A., Toth, P.: The vehicle routing problem. In: Christofides, N., Mingozzi, A., Toth, P., Sandi, C. (eds.) Combinatorial Optimization, pp. 315–338. Wiley, Chichester (1979)
15. Tarantilis, C.D., Kiranoudis, C.T.: Using the Vehicle Routing Problem for the Transportation of Hazardous Materials. Oper. Res. An. Int. J. **1**(1), 67–78 (2001)

R*-Tree

HANS-PETER KRIEGEL, PETER KUNATH,
MATTHIAS RENZ
Department of Computer Science, Ludwig-Maximilians-University of Munich, Munich, Germany

Synonyms

Spatial index structure; Spatial access method; R-tree; TPR-trees; Point query; Window query; Range query; Directory rectangles; Storage utilization; Reinsert, forced

Definition

The R*-tree, an improvement of the R-tree, is one of the most popular access methods for points and rectangles. This is achieved by modifying the insert and split algorithms of the original R-tree which is based on the heuristic optimization of the area of the enclosing rectangle in each inner node. The R*-tree incorporates a combined optimization of area, margin and overlap of each enclosing rectangle in the directory. From a practical point of view, the R*-tree is very attractive because of the following two reasons. It efficiently supports point and spatial data at the same time and its implementation cost is only slightly higher than that of other R-tree variants.

Historical Background

The R*-tree, proposed in 1990, is one of the most prominent representatives of the R-tree family. One major reason for using R-tree based index structures is the requirement to index not only point data but also extended spatial data, and R-tree-based index structures are well suited for both types of data. In contrast to most other index structures (such as kdB-trees [1], grid files [2], and their variants, see e. g., [3]), R-tree-based index structures do not need point transformations to store spatial data and, therefore, provide better spatial clustering. The R*-tree differs from the R-tree in its split strategy which is based on a heuristic optimization. It constitutes the foundation of several modern spatial access methods including the X-tree [4] or the TPR-tree [5], specialized access methods that cope with high-dimensional data and moving objects.

Scientific Fundamentals

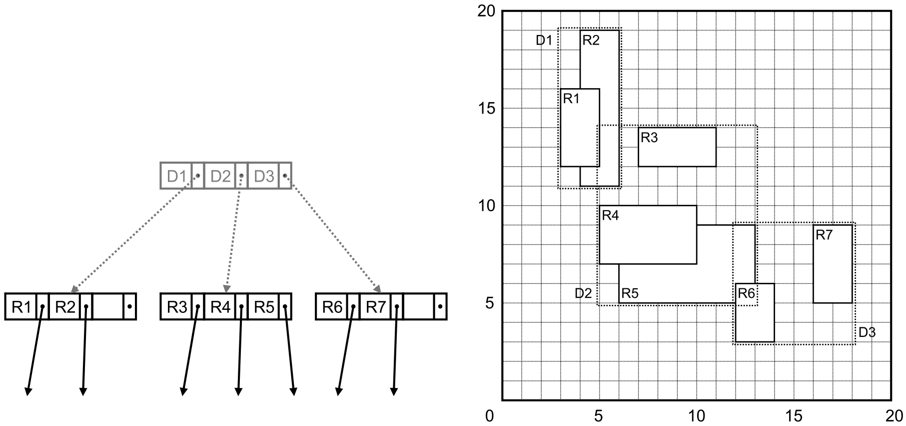
The R*-tree proposed in [6] is an extension of the R-tree which was designed to overcome certain drawbacks of the R-tree. An example of the R*-tree structure and the data space with the corresponding bounding rectangles is depicted in Fig. 1.

Optimization Criteria for R-Trees

For a set of given data objects, an R-tree dynamically constructs bounding boxes from subsets between m and M rectangles. This is done in a way that efficiently supports retrieval operations such as point- and window-queries of arbitrary size. The known parameters of good retrieval performances affect each other in a very complex way, such that it is impossible to optimize one of them without influencing other parameters. This may then lead to a deterioration of the overall performance. Moreover, since the data rectangles may have very different size and shape, and the directory rectangles grow and shrink dynamically, the success of methods which optimize only one parameter is very unlikely.

In the following text, some of the parameters which are essential for the retrieval performance are considered. Furthermore, interdependencies between different parameters and optimization criteria are analyzed.

(O1) The area covered by a directory rectangle should be minimized, i. e., the area covered by the bounding rectangle, but not covered by the enclosed rectangles,



Structure of the R*-tree with inner nodes (light color) and leaf nodes (dark color).

Data space with directory rectangles D1 – D3 (dashed lines) and data rectangles R1 – R7 (solid lines).

R*-Tree, Figure 1 R*-tree example

should be minimized. This area is often referred to as *dead space*. This improves the performance since decisions regarding which paths have to be traversed can be made on higher levels.

(O2) The overlap between directory rectangles should be minimized. This also decreases the number of paths to be traversed.

(O3) The margin of a directory rectangle should be minimized. Here, the margin is the sum of the lengths of the edges of a rectangle. Assuming a fixed area, the rectangular object with the smallest margin is a square. Thus, minimizing the margin instead of the area will result in directory rectangles that are more quadratic shaped. Queries with large quadratic query rectangles especially profit from this optimization. More importantly, the minimization of the margin improves the overall structure. Since quadratic objects can be packed easier, the bounding boxes of a level will form smaller directory rectangles on the level above. Thus, clustering rectangles into bounding boxes with only little variance of the length of the edges will reduce the area of the directory rectangles.

(O4) The storage utilization should be optimized. Higher storage utilization generally reduces the query cost as the height of the tree is kept low. Queries with a low selectivity especially benefit from a good storage utilization because the number of visited nodes for such queries is usually higher than that for queries with a high selectivity.

Unfortunately, these four criteria are not necessarily in accordance with each other. Keeping the area and overlap of a directory rectangle small requires more freedom in the number of rectangles stored in one node. Minimizing these parameters usually comes at the cost of decreasing the storage utilization. Moreover, when fulfilling (O1) or (O2), more freedom in choosing the shape is necessary. Thus, the rectangles will be less quadratic. When focusing on (O1), the overlap between directory rectangles may be affected in a positive way since the data space covered by the directory rectangles, in particular the *dead space*, is reduced. As for every geometric optimization, minimizing the margins also leads to a reduced storage utilization. However, since a more quadratic shape of the directory rectangles results in better packing, it is easier to maintain high storage utilization. Obviously, the performance of queries with sufficiently low selectivity will be affected more by the storage utilization than by the parameters of (O1)–(O3).

Choosing the Subtree

To solve the problem of choosing an appropriate insertion path, previous R-tree versions take only the area into consideration. The authors of the R*-tree tested the parameters area, margin, and overlap in different combinations. The overlap of an entry is defined as follows. Let $E_1, \ldots, E_p$ be the entries in the current node. Then,

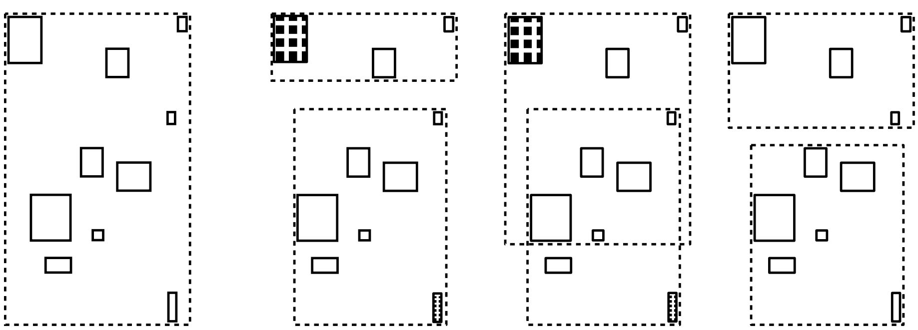
$$\mathrm{overlap}(E_k) = \sum_{i=1, i \neq k}^{p} \mathrm{area}(E_i \cap E_k), \; 1 < k < p.$$

Figure 2 describes the algorithm which performed best in the experimental evaluation conducted by the R*-tree authors. For choosing the best non-leaf node, alternative methods did not outperform Guttman's original algorithm. For the leaf nodes, minimizing the overlap performed slightly better.

In this version, the CPU cost to determine the overlap are quadratic in the number of node entries, because for each entry, the overlap with all other entries of the node has to be calculated. However, for large node sizes, one can reduce

```
ChooseSubtree(N: Node)
CS1  Choose N to be the root
CS2  IF N is a leaf
           return N
       ELSE
           IF the childpointers in N point to leaves
               // Determine minimum overlap cost
               Choose the entry in N whose rectangle needs
               least overlap enlargement to include the new data.
               Resolve ties by choosing the entry whose rectangle
               needs least area enlargement.
               If there is still a tie, choose the entry with
               the rectangle that has the smallest area.
           ELSE // the childpointers in N do not point to leaves
               // Determine minimum area cost
               Choose the entry in N whose rectangle needs least
               area enlargement to include the new data rectangle.
               Resolve ties by choosing the entry with the
               rectangle that has the smallest area.
           END
       END
CS3      Set N to be the childnode pointed to by the
         childpointer of the chosen entry and repeat from CS2
```

R*-Tree, Figure 2 The algorithm for finding the most appropriate subtree



Overfilled node | Split of the quadratic R-tree, m = 30% | Split of the quadratic R-tree, m = 40% | Split of the R*-tree, m = 40%

R*-Tree, Figure 3 Different partitioning variants

the number of entries for which the calculation has to be done. The idea is that for very distant rectangles, the probability to yield the minimum overlap is very small. Thus, in order to reduce the CPU cost, the first part of CS2 can be modified as follows:

> // Determine the minimum overlap cost (fast version)
> Sort the rectangles in node *N* in increasing order of their area enlargement needed to include the new data rectangle. Let *A* be the group of the first *p* entries.
> From the entries in *A*, consider all entries in *N* and choose the entry whose rectangle needs least overlap enlargement. Resolve ties as described above.

The R*-tree authors showed that this modified version leads to almost no reduction of retrieval performance when using two-dimensional data and setting the *p* parameter to 32. Nevertheless, the CPU costs remain higher than in the R-tree version of `ChooseSubtree`. However, the number of disk accesses is reduced for the exact match query preceding each insertion and is reduced for the `ChooseSubtree` algorithm itself. In particular, highly selective window queries on data sets consisting of non-uniformly distributed small rectangles or points benefit from the `ChooseSubtree` optimization. For other data sets, the performance of Guttman's algorithm is usually similar to this one.

Splitting a Node

For determining a good split of a directory node, it is necessary to find an adequate partitioning of the MBRs in the node into two subsets. Figure 3 illustrates typical problems that arise when using an inappropriate partitioning of the entries of a node. The example shows that the split strategy of the R*-tree leads to a more effective partitioning than that of the R-tree. The depicted split results are based on the quadratic split of the R-tree for a varying mini-

Split
S1 Invoke ChooseSplitAxis to determine the axis perpendicular to which the split is performed.
S2 Invoke ChooseSplitIndex to determine the best distribution into two groups along that axis.
S3 Distribute the entries into two groups.

ChooseSplitAxis
CSA1 For each axis: Sort the entries by the lower value, then by the upper value of their rectangles and determine all distributions.
Compute S, the sum of all margin-values of the different distributions.
CSA2 Choose the axis with the minimum S as the split axis.

ChooseSplitIndex
CSI1 Along the chosen split axis, choose the distribution with the minimum overlap-value. Resolve ties by choosing the distribution having the minimum area-value.

R*-Tree, Figure 4 The R*-tree Split algorithm

mum number of entries m relative to M. The result is either a split with uneven distribution of the entries, reducing the storage utilization (first partitioning), or a split with much overlap (second partitioning). For comparison, the third partitioning illustrates the results from the R*-tree split.

The R*-tree uses the following method to find good splits which is depicted in Fig. 4.

In a first step (S1), the split axis has to be chosen. This is carried out using the function *ChooseSplitAxis* in the following way. Along each axis, the entries are first sorted by the lower value of their rectangles, then sorted by the upper value of their rectangles. For each sort $M-2m+2$ distributions of the $M+1$ entries into two groups are determined, where the k-th distribution $(k=1,\dots,(M-2m+2))$ is determined as follows. The first group contains the first $(m-1)+k$ entries, the second group contains the remaining entries. For each of the $M–2m+2$ distributions, the *margin-value* is determined by summarizing the margin length of the two minimum bounding boxes of both distributions. Finally, the axis which yields the minimum *margin-value* is chosen as a split axis.

In the next step (S2), an adequate partitioning of the entries along the split axis determined in the previous step has to be found. This is done by the function *ChooseSplitIndex*. This time, the *overlap-value* for each of the $2\cdot(M-2m+2)$ distributions is considered where the *overlap-value* denotes the size of the area of the overlap between the two minimum bounding boxes of both partitions. Here, ties are resolved by choosing the distribution with the minimum *area-value* denoting the sum of the size of the areas covered by both minimum bounding boxes of both partitions.

Note that each of the parameters, *margin-value*, *overlap-value* and *area-value*, potentially may be chosen for the determination of the split axis and the final distribution in an arbitrary sequence. In the present experiments, the proposed constellation, i. e., *margin-value* for the determination of the split axis and *overlap-value* and *area-value* for the final choice of the partitioning, has shown the best overall performance.

As experiments with several values of M have shown, $m = 40\%$ yields the best performance. For each axis (dimension), the entries have to be sorted twice, requiring $O(M \log M)$ time. As an experimental cost analysis showed, this needs about half of the cost of the split. The remaining split cost is spent computing the margin of the $2\cdot(M-2m+2)$ distributions.

Forced Reinsert

Both the R-tree and the R*-tree are nondeterministic in allocating the entries to the nodes, i. e., different sequences of insertions will generate different trees. Obviously, the retrieval performance of the R-tree can suffer from its old entries. Data rectangles inserted during the early growth of the structure may have introduced directory rectangles which are not suitable to guarantee a good retrieval performance in the current situation. A very local reorganization of the directory rectangles is performed during a split. However, this is rather poor and it is thus desirable to have a more powerful and less local instrument to reorganize the structure.

The discussed problem would be maintained or even worsened if underfilled nodes, resulting from the deletion of records, would be merged under the old parent. Thus, the approach of treating underfilled nodes in an R-tree is to delete the node and to reinsert the orphaned entries in the corresponding level [7]. This way, the ChooseSubtree algorithm has a new chance of distributing entries into different nodes. Therefore, randomly deleting parts of the data and then reinserting it seems to be a very simple way of tuning existing R-tree data files. However, this is a static

InsertData
ID1 Invoke Insert starting with the leaf level as a parameter to insert a new data rectangle.

Insert
I1 Invoke ChooseSubtree with the level as a parameter to find an appropriate node N, in which to place the new entry E.
I2 If N has less then M entries, accommodate E in N.
If N has M entries, invoke OverflowTreatment with the level of N as a parameter (for reinsertion or split).
I3 If OverflowTreatment was called and a split was performed, propagate OverflowTreatment upwards if necessary.
If OverflowTreatment caused a split of the root, create a new root.
I4 Adjust all covering rectangles in the insertion path such that they are minimum bounding boxes enclosing their children rectangles.

OverflowTreatment
OT1 If the level is not the root level and this is the first call of OverflowTreatment in the given level during the insertion of one data rectangle, then invole Reinsert, else invoke Split.

Reinsert
RI1 For all $M+1$ entries of a node N compute the distance between the centers of their rectangles and the center of the bounding rectangle of N.
RI2 Sort the entries in decreasing order of their distances computed in RI1.
RI3 Remove the first p entries from N and adjust the bounding rectangle of N.
RI4 In the sort defined in RI2, starting with the maximum distance (= far reinsert) or minimum distance (= close reinsert), invoke Insert to reinsert the entries.

R*-Tree, Figure 5 The R*-tree Reinsert algorithm

situation, and for nearly static data files, the pack algorithm [8] is a more sophisticated approach.
In order to achieve dynamic reorganizations, the R*-tree forces entries to be reinserted during the insertion routine. The algorithm `Reinsert` in Fig. 5 is based on the ability of the insert routine to insert entries on every level of the tree, as already required by the deletion algorithm [7].
If a new data rectangle is inserted, the first overflow treatment on each level will be reinsertion of p entries, where p is a percentage value to be optimized. This may cause a split in the node which caused the overflow if all entries are reinserted in the same location. Otherwise, splits may occur in one or more of the other nodes, however, in many situations splits are completely prevented. The parameter p can be varied independently for leaf nodes and non-leaf nodes as part of performance tuning. Experiments have shown that $p = 30\%$ of M for leaf nodes as well as non-leaf nodes yield the best performance. Furthermore, experiments demonstrated that close Reinsert outperforms far Reinsert. Close reinsert prefers the node which included the entries before, and this is intended, because its enclosing rectangle was reduced in size. Thus, this node has lower probability to be selected by ChooseSubtree again.

In summary, the following properties apply to the concept of Forced Reinsert.

- Forced Reinsert relocates entries between neighboring nodes and thus decreases the overlap.
- As a side effect, storage utilization is improved.
- Due to more restructuring, less splits occur.
- Since the outer rectangles of a node are reinserted, the shape of the directory rectangles will be more quadratic.
- Higher CPU cost due to the need to call the insertion routine more often is alleviated by having to perform less splits.
- Experimental evaluations show that even with Forced Reinsert, the average insertion cost of the R*-tree is lower than that for the other R-tree variants.

Key Applications

A spatial object is regarded as a distinct entity occupying an individual location in a one- or multidimensional data space. Furthermore, it may be extended along some (or all) dimensions. A complex spatial object of the real world can be considered as a collection of individual, two- or three-dimensional components, where each component potentially represents a complex and intricate geometric shape. Examples of such complex objects are geographical regions, or parts of a car or an airplane.

Geographic Information Systems (GIS)

Geographic Information Systems, abbreviated as GIS, are computer-based systems that have been developed and designed specifically to handle geographic information. They are designed to allow the user to capture spatial information, its storage, analysis, manipulation and the production of maps as outputs. Since spatial information can be efficiently handled by means of spatial access methods like the R*-tree, these systems are able to hold an enormous range and quantity of information.

Digital Mock-up (DMU)

In CAD databases, each instance of a part occupies a specific region in the two- or three-dimensional product space. Together, all parts of a given product version and its variants represent a virtual prototype of the constructed geometry. One of the most important tasks for DMU are collision detection and proximity queries that can be efficiently supported by the R*-tree.

Multidimensional Feature Vectors

While data sets usually are organized with respect to primary keys, the remaining entries often resemble the interesting part of the data set. A set of numerical (and also certain categorical) features often fulfills the properties of vector spaces, and thus, database objects can be considered as points in space. The R*-tree is one of the most frequently used access methods for objects in multidimensional vector spaces. It can manage point objects as well as spatially extended objects. Due to its good clustering properties, it is very suitable for proximity queries like distance range queries and nearest neighbor queries that are often used for similarity search.

Future Directions

The R*-tree can be efficiently used as an access method in database systems organizing both multidimensional point data and spatial data. Since modern access methods have to cope with more complex object representations such as high-dimensional data, dynamic objects, or uncertain data, variants or extensions of the R*-tree may be more suitable. Examples of such variants are the X-tree (for high-dimensional data), the TPR-tree (for dynamic objects), and the Gauss-tree (for uncertain data).

Cross References

► Indexing, High Dimensional
► Indexing, Hilbert R-tree, Spatial Indexing, Multimedia Indexing
► Indexing, X-Tree
► R-Trees – A Dynamic Index Structure for Spatial Searching
► Vector Data

Recommended Reading

1. Robinson, J.T.: The K-D-B-tree: a search structure for large multidimensional dynamic indexes. In: Proc. of the 1981 ACM SIGMOD international conference on Management of data (SIGMOD'81), Ann Arbor, Michigan (1981)
2. Nievergelt, J., Hinterberger, H., Sevcik, K.C.: The grid file: An adaptable, symmetric multikey file structure. ACM Trans Database Syst **9**(1):38–71 (1984)
3. Seeger, B., Kriegel, H.-P.: The buddy tree: an efficient and robust access method for spatial data bases. In: Proceedings of 16th international conference on Very large databases (VLDB'90), Brisbane, Australia (1990) pp. 590–601
4. Berchtold, S., Keim, D.A., Kriegel, H.-P: The X-tree: An index structure for high-dimensional data. In: Proc. 22nd Int. Conf. on Very Large Data Bases (VLDB'96), Bombay, India (1996) pp. 28–39
5. Saltenis, S., Jensen, C.S., Leutenegger, S.T., Lopez, M.A.: Indexing the positions of continuously moving objects. In: SIGMOD Conference, (2000) pp. 331–342
6. Beckmann, N., Kriegel, H.-P., Schneider, R., Seeger, B.: The R*-tree: An efficient and robust access method for points and rectangles. In: Proceedings of the International Conference on Management of Data (SIGMOD'90), Atlantic City, NJ (1990) pp. 322–331
7. Guttman, A.: R-trees: a dynamic index structure for spatial searching. In: Proc. of the 1984 ACM SIGMOD international conference on Management of data (SIGMOD'84), Boston, Massachusetts (1984) pp. 47–57
8. Roussopoulos, N., Leifker, D.: Direct spatial search on pictorial databases using packed R-trees. In: Proc. of the 1985 ACM SIGMOD international conference on Management of data (SIGMOD'85), Austin, Texas (1985) pp. 17–31

R-Tree

► Indexing Schemes for Multi-dimensional Moving Objects
► Indexing, X-Tree
► PostGIS
► R*-tree
► R-Trees – A Dynamic Index Structure for Spatial Searching

R-Tree, Multi-Version

► Movement Patterns in Spatio-temporal Data

R-Trees – A Dynamic Index Structure for Spatial Searching

MARIOS HADJIELEFTHERIOU[1],
YANNIS MANOLOPOULOS[2], YANNIS THEODORIDIS[3],
VASSILIS J. TSOTRAS[4]
[1] AT&T Labs, Inc., Florham Park, NJ, USA
[2] Department of Informatics, Aristotle University, Thessaloniki, Greece
[3] Department of Informatics, University of Piraeus, Piraeus, Greece
[4] University of California-Riverside, Riverside, CA, USA

Synonyms

R-tree

Definition

One of the most influential access methods in the area of Spatial Data Management is the R-tree structure proposed by Guttman in 1984 [8]. It is a hierarchical data structure based on B^+-trees, used for the dynamic organization of a set of d-dimensional geometric objects. The original R-tree was designed for efficiently retrieving geometric objects contained within a given query range. Every object in the R-tree is represented by a minimum bounding d-dimensional rectangle (for simplicity, MBRs in the sequel). Data objects are grouped into larger MBRs forming the *leaf nodes* of the tree. Leaf nodes are grouped into larger *internal nodes*. The process continues recursively until the last group of nodes that form the root of the tree. The root represents an MBR that encloses all objects and nodes indexed by the tree, and each node corresponds to the MBR that bounds its children (cf. Fig. 1). A range query can be answered efficiently by traversing the tree starting from the root and ending at the leaves, accessing only nodes whose MBRs intersect with the query range. At the leaf level of the tree, the actual geometric objects are retrieved and tested for true containment in the query. Several variations of the original structure have been proposed to provide more efficient access, handle objects in high-dimensional spaces, support concurrent accesses, support I/O and CPU parallelism, support efficient bulk loading, and several other types of spatial and spatio-temporal queries, like nearest neighbors, spatial joins, and more.

Historical Background

The 1980s were a period of wide acceptance of relational systems in the market, but at the same time it became apparent that the relational model was not adequate to host new emerging applications. Multimedia, CAD/CAM, geographical, medical and scientific applications are just some examples, in which the relational model had been proven to behave poorly. Thus, the object-oriented model and the object-relational model were proposed. One of the reasons for the shortcoming of the relational systems was their inability to handle the new kinds of data with B-trees. More specifically, B-trees were designed to handle alphanumeric (i. e., one-dimensional) data, like integers, characters, and strings, where an ordering relation can be defined. In light of this development, entirely novel access methods were proposed, evaluated, compared, and established. One of these structures, the R-tree, was proposed by Guttman in 1984, aimed to handle geometrical data, such as points, line segments, surfaces, volumes, and hyper volumes in high-dimensional spaces [8]. R-trees were treated in the literature in much the same way as B-trees. In particular, many improving variations have been proposed for various instances and environments, several novel operations have been developed, and new cost models have been suggested.

It seems that due to modern demanding applications and after academia has paved the way, the industry recently recognized the use and necessity of R-trees. Thus, R-trees are adopted as an additional access method to handle multi-dimensional data. Nowadays, spatial databases and geographical information systems have been established as a mature field, spatio-temporal databases and manipulation of moving points and trajectories are being studied extensively, and finally image and multimedia databases able to handle new kinds of data, such as images, voice, music, or video, are being designed and developed. An application in all these cases should rely on R-trees as a necessary tool for data storage and retrieval. R-tree applications cover a wide spectrum, from spatial and temporal to image and video (multimedia) databases. The initial application that motivated Guttman in his pioneering research was VLSI design (i. e., how to efficiently answer whether a space is already covered by a chip). Gradually, handling rectangles quickly found applications in geographical and, in general, spatial data, including GIS (buildings, rivers, cities, etc.), image or video/audio retrieval systems (similarity of objects in either original space or high-dimensional feature space), time series and chronological databases (time intervals are just 1D objects), and so on [18].

Scientific Fundamentals

The Original R-Tree

The R-tree comprises a generalization of the B^+-tree structure for multiple dimensions. An R-tree of order (m, M) has the following characteristics:

1. The root node of the tree contains at least two entries, unless it is a leaf (in this case, it may contain zero or a single entry).
2. Internal nodes can store between $m \le M/2$ and M child entries. Each entry is of the form (*p, mbr*), where *p* is a pointer to a children node and *mbr* is the MBR that spatially encloses all entries contained in the sub-tree rooted at this child.
3. Each leaf node (unless it is the root) can store between $m \le M/2$ and M entries. Each entry is of the form *oid*, *mbr*), where *oid* is an object identifier and *mbr* is the MBR that spatially encloses this object.
4. The R-tree is a height-balanced structure, i. e., all leaves appear at the same level of the tree.

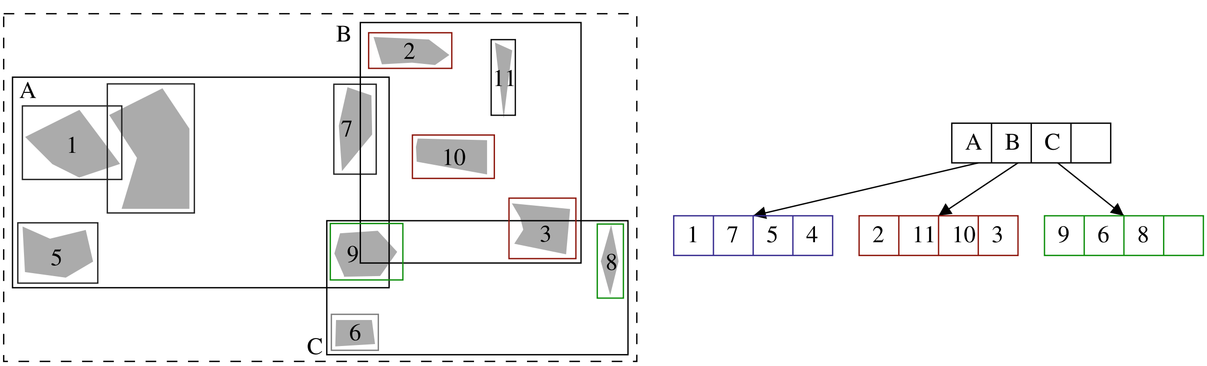
Let an R-tree store N data objects. The maximum possible height h of the tree is $h_{\max} = \lceil \log_m N \rceil - 1$. The maximum number of nodes (by assuming that each node contains the minimum allowed number of entries) is $\sum_{i=1}^{h_{\max}} \lceil N/m^i \rceil = \lceil N/m \rceil + \lceil N/m^2 \rceil + \ldots + 1$.

Figure 1 shows a set of geometric objects, their enclosing MBRs, and a grouping into larger MBRs forming the index levels of the R-tree (assuming that $M = 4$ and $m = 2$). The actual hierarchy of the R-tree is shown on the right. It is evident that several R-trees can represent the same set of data rectangles, depending on the grouping of data objects into leaves, and internal nodes into larger nodes. Hence, which R-tree is constructed is determined by the insertion and/or deletion algorithm used.

Notice also that node MBRs on the same level of the tree might overlap each other. Besides, in a geometric sense a given MBRs may be contained in multiple nodes, but from an R-tree perspective it is associated with only one node of the tree. This means that a spatial search may have to traverse multiple paths of the tree before confirming the existence of a given MBR. Also, the representation of geometric objects with MBRs may result in false alarms. To resolve false alarms, the actual objects must be examined. Therefore, the R-tree plays the role of a filtering mechanism to reduce the costly direct examination of geometric objects, and limit the search space.

Given a query rectangle Q *a range search or a window query* (i. e., retrieving all data objects that intersect with Q) is performed by traversing the tree starting from the root and ending at the leaves, retrieving only nodes that intersect with Q. The algorithm is shown in Fig. 2. For a node entry *e*, *e.mbr* denotes the child MBR and *e.p* the child pointer to the next level. If the node is a leaf, then *e.p* is the object identifier (*oid*).

Note that the objects returned by the RangeSearch procedure are potential candidate answers. The actual geometric



R-Trees – A Dynamic Index Structure for Spatial Searching, Figure 1 The hierarchy of an R-tree. Data objects are represented by their enclosing MBRs, which are grouped into larger nodes hierarchically until the root node of the tree

```
Algorithm RangeSearch (TypeNode RN, TypeRegion Q)
/* Finds all rectangles that are stored in an R-tree with root node RN, which
intersect with a query rectangle Q. Answers are stored in set A */

if RN is not a leaf node
    examine each entry e of RN to find e.mbr that intersect Q
    foreach such entry e call RangeSearch(e.p,Q)
else // RN is a leaf node
    examine each entry e to find e.mbr that intersects Q
    add these entries to answer set A
endif
```

R-Trees – A Dynamic Index Structure for Spatial Searching, Figure 2 The R-tree range search algorithm

Algorithm Insert (TypeEntry E, TypeNode RN)
/* Inserts a new entry E in an R-tree with root node RN */

1. Traverse the tree from root RN to the appropriate leaf. At each level,
   select the node, L, whose MBR will require the minimum area enlargement
   to cover $E.mbr$
2. In case of ties, select the node whose MBR has
   the minimum area
3. **if** the selected leaf L can accommodate E
4. Insert E into L
5. Update all MBRs in the path from the root to L,
   so that all of them cover $E.mbr$
6. **else** // L is already full
7. Let $\mathcal{E}$ be the set consisting of all L's entries and the new entry E
   Select as seeds two entries $e_1, e_2 \in \mathcal{E}$, where the distance between
   e_1 and e_2 is the maximum among all other pairs of entries from $\mathcal{E}$
   Form two nodes, L_1 and L_2, where the first contains e_1 and the second e_2
8. Examine the remaining members of $\mathcal{E}$ one by one and assign them
   to L_1 or L_2, depending on which of the MBRs of these nodes
   will require the minimum area enlargement so as to cover this entry
9. **if** a tie occurs
10. Assign the entry to the node whose MBR has the smaller area
11. **endif**
12. **if** a tie occurs again
13. Assign the entry to the node that contains the smaller number of entries
14. **endif**
15. **if** during the assignment of entries, there remain λ entries to be assigned
    and the one node contains $m - \lambda$ entries
16. Assign all the remaining entries to this node without considering
    the aforementioned criteria
    /* so that the node will contain at least m entries */
17. **endif**
18. Update the MBRs of nodes that are in the path from root to L, so as to
    cover L_1 and accommodate L_2
19. Perform splits at the upper levels if necessary
20. In case the root has to be split, create a new root
21. Increase the height of the tree by one
22. **endif**

R-Trees – A Dynamic Index Structure for Spatial Searching, Figure 3 The R-tree insertion algorithm

objects intersected by the query rectangle have to be identified using a refinement step by retrieving the objects and testing for true intersection.

R-trees are dynamic data structures, i. e., global reorganization is not required to handle insertions or deletions. Insertions are handled similarly to insertions in a B^+-tree. In particular, the R-tree is traversed to locate an appropriate leaf to accommodate the new entry. The entry is inserted in that leaf and all nodes in the path from the leaf to the root are updated accordingly. In case the leaf cannot accommodate the new entry because it already contains M entries, it is split into two new nodes. The algorithm for inserting a new data rectangle in an R-tree is presented in Fig. 3.

Splitting in R-trees differs from that of the B^+-tree. Given that R-tree nodes are allowed to overlap, the objective of a split algorithm is to minimize the probability of retrieving both new nodes (L_1 and L_2) for the same query. The splitting criteria introduced by Guttman are the following:

Linear Split. Choose the two objects lying the furthest apart from each other as seeds for the two new nodes. Then consider each remaining object in a random order and assign it to the node requiring the smallest enlargement of its respective MBR. The linear split algorithm tries to minimizing the total area of the two new nodes.

Quadratic Split. Choose two objects as seeds for the two nodes. The seeds, if grouped together create as much dead space as possible (*dead space* is the area covered by the MBR but not by the enclosed children). Then, until there are no remaining objects, choose the object

Algorithm Delete (TypeEntry E, TypeNode RN)
/* Deletes an entry E from an R-tree with root node RN */

```
if RN is a leaf node
    search all entries of RN to find E.mbr
else // RN is an internal node
    Find all entries of RN that cover E.mbr
    Follow the corresponding subtrees until the leaf L that contains E is found
    Remove E from L
endif
Call algorithm CondenseTree(L) /* Figure 5 */
if the root has only one child /* and it is not a leaf */
    Remove the root
    Set as new root its only child
endif
```

R-Trees – A Dynamic Index Structure for Spatial Searching, Figure 4 The R-tree deletion algorithm

that maximizes the difference of dead space if assigned to each of the two nodes, and insert it in the node that requires the least enlargement of its MBR.

Exponential Split. All possible groupings are exhaustively tested and the best is chosen with respect to the minimization of the MBR enlargement.

Regarding the deletion of an entry from an R-tree, it is performed with the algorithm given in Fig. 4. Notice that the R-tree is using re-insertions instead of merging sibling nodes to handle entries contained in nodes that underflow (i. e., nodes that end up with less than m children). Reinsertion achieves the same result as merging. Additionally, the algorithm for insertion is reused. Also, most of the nodes accessed during reinsertion are available in buffer memory, because they were retrieved during searching for the deleted entry. The standard deletion technique is compared with alternative techniques in [19].

R-Tree Variants

The original R-tree has two important disadvantages that motivated the study of more efficient variations:

1. The execution of a point location query in an R-tree may lead to the investigation of several paths from the root to the leaf level. This characteristic may lead to performance deterioration, specifically when the overlap of the MBRs is significant.
2. A few large rectangles may increase the degree of overlap significantly, leading to performance degradation during range query execution, due to empty space.

The R$^+$-Tree R$^+$-trees [23] were proposed as a structure that avoids visiting multiple paths during point location queries, and thus leading to improved retrieval performance. The R$^+$-tree avoids MBR overlapping of internal nodes at the same level of the tree, by using the clipping technique. Inserted objects and nodes are divided in two or more MBRs if needed. This means that a specific object *oid* may be duplicated and stored redundantly in several nodes.

The algorithm for range query processing is similar to the one used for R-trees. The only difference is that duplicate elimination is necessary to avoid reporting an object more than once. However, insertion, deletion, and node splitting algorithms are different due to the clipping technique. In order to insert a new entry E, the insertion algorithm starts from the root and determines the MBRs that intersect *E.mbr*. Then *E.mbr* is clipped and the procedure is recursively applied for the corresponding sub-trees. During the execution of the insertion algorithm a node may become full. To handle this situation, a node splitting mechanism is required as in the R-tree case. The main difference between the R$^+$-tree splitting algorithm and that of the R-tree is that downward propagation may be necessary due to object redundancy, in addition to the upward propagation. At the same time, another side effect of clipping is that during insertions, an MBR augmentation may lead to a series of update operations in a chain reaction type. Also, under certain circumstances, the structure may lead to a deadlock, as, for example, when a split has to take place at a node with M+1 rectangles, where every rectangle is enclosed into another. The fact that multiple copies of an object's MBR may be stored in several leaf nodes has a direct impact on the deletion algorithm. All copies of an object's MBR must be removed from the corresponding leaf nodes. Evidently, an increased number of deletions may reduce storage utilization significantly. Therefore, appropriate reorganization must be performed to handle underutilized tree nodes as well.

The R*-Tree R*-trees [2] are widely accepted in the literature as a prevailing performance-wise structure that is

Algorithm CondenseTree (TypeNode L)
/* Given is the leaf L from which an entry E has been deleted, if after the deletion of E, L has fewer than m entries, remove entirely leaf L and reinsert all remaining entries. Updates are propagated upwards and the MBRs in the path from the root to L are modified (possibly become smaller) */

```
1.  Set X = L
2.  Let 𝒩 be the set of nodes that are going to be removed from
    the tree (initially, 𝒩 is empty)
3.  while X is not the root
4.      Let Parent_X be the father node of X
5.      Let E_X be the entry of Parent_X that corresponds to X
6.      if X contains less than m entries
7.          Remove E_X from Parent_X
8.          Insert X into 𝒩
9.      endif
10.     if X has not been removed
11.         Adjust its corresponding MBR E_X.mbr, so as to enclose
            all rectangles in X /* E_X.mbr may become smaller */
12.     endif
13.     Set X = Parent_X
14. endwhile
15. Reinsert all the entries of nodes that are in the set 𝒩
```

R-Trees – A Dynamic Index Structure for Spatial Searching, Figure 5 The R-tree condense algorithm

often used as a basis for performance comparisons. As already discussed, the R-tree is based solely on the area minimization of node MBRs during splitting. On the other hand, the R*-tree examines several other splitting criteria, which intuitively are expected to improve the performance during query processing. The criteria considered are the following:

Minimization of the area covered by each MBR. This criterion aims to minimize the dead space and consequently reduce the number of paths that need to be traversed during query processing. This is the single criterion that is also examined by the R-tree.

Minimization of overlap between MBRs. The larger the overlapping, the larger the expected number of paths traversed by a query.

Minimization of MBR margins (perimeters). Minimization of the perimeter has a direct effect on the shape of the MBR. Regular quadrilaterals (squares) have the smallest perimeter between rectangles with equal area (as opposed to elongated rectangles). Regular shapes help group MBRs more compactly and reduce empty space.

Maximization of storage utilization. When utilization is low, more nodes tend to be examined during query processing, especially for larger queries where a significant portion of the entries intersect with the query. Moreover, the tree height increases with decreasing node utilization.

The R*-tree follows an engineering approach to find the best possible combinations of the aforementioned criteria. This heuristic approach is necessary, because the criteria can become contradictory. For instance, to keep both the area and overlap low, one can reduce the minimum number of entries allowed within a node. Although, this has direct impact on storage utilization. Also, by minimizing the margins in order to have more regular shapes, node overlapping may be increased.

For the insertion of a new entry e, the R*-tree decides which branch to follow by choosing the entry whose MBR needs the *least area enlargement* to cover *e.mbr*. For leaf nodes, the R*-tree considers node *overlapping minimization* instead. For leaves that are full, the R*-tree does not immediately resort to node splitting. Instead, it picks a fraction of the entries from the chosen leaf and reinserts them. The set of entries to be reinserted are those whose centroid distances from node centroid are among the largest 30% (i. e., this is a heuristic to detect and discard the furthest entries). The reinsertion algorithm achieves a kind of tree re-balancing that significantly improves performance during query processing. However, reinsertion is a costly operation. Therefore, only one application of reinsertion is permitted for each level of the tree. When overflow cannot be handled by reinsertion, node splitting is performed. The split algorithm consists of two steps. The first step decides a split axis among all dimensions. The split axis is the one with the smallest overall perimeter. When the split axis is selected, the split algorithm sorts the entries (according to their lower or upper boundaries) on the selected dimension and examines all possible divisions. The final division is the one that has the minimum overlap between the MBRs of the resulting nodes. The cost of the split algorithm of the R^*-tree is computed as follows. Let M and m be the maximum and minimum allowed number of entries, respectively. The entries are sorted once for every dimension, with cost $O(M \log M)$. For each axis, the margin of 2 x 2 x ($M - 2m + 2$) rectangles and the overlap of 2 x ($M - 2m + 2$) divisions is calculated. The R^*-tree does not use any specialized deletion algorithm. Instead, deletion in the R^*-tree is performed with the deletion algorithm of the original R-tree.

Other Variants The Hilbert R-tree [13] is a hybrid structure. It is a B^+-tree with geometrical objects being characterized by the Hilbert value of their centroid. The structure is based on the Hilbert space-filling curve which has been shown to preserve the proximity of spatial objects very well.

Ang and Tan in [1] have proposed a linear algorithm to distribute the objects of an overflowing node in two sets. The primary criterion of this algorithm is to distribute the objects between the two nodes as evenly as possible, whereas the second criterion is the minimization of the overlapping between them. Finally, the third criterion is the minimization of the total coverage.

Garcia et al. [7] proposed a new optimal polynomial splitting algorithm $O(n^d)$, where d is the space dimensionality and $n = M + 1$ is the number of entries of the node that overflows. For n rectangles the number of possible bi-partitions is exponential in n. Each bi-partition is characterized by a pair of MBRs, one for each set of rectangles in each partition. The key issue, however, is that a large number of candidate bi-partitions share the same pair of MBRs. This happens when rectangles that do not participate in the formation of the new enclosing MBRs are exchanged. The authors show that if the cost function used depends only on the characteristics of the MBRs, then the number of different partitionings is polynomial.

More recently, Schreck and Chen [22] proposed an insertion heuristic to improve the shape of the R-tree so that the tree achieves a more elegant shape, with a smaller number of nodes and better storage utilization. In particular, this technique considers how to redistribute data among neighboring nodes, so as to reduce the total number of created nodes. This approach is motivated by the fact that if an overflowing entry could be accommodated by another node the split could be prevented. In this case, a split is performed only when all nodes are completely full.

Huang et al. proposed Compact R-trees, a dynamic R-tree version with optimal space overhead [11]. The motivation behind the proposed approach is that R-trees, R^+-trees, and R^*-trees suffer from the storage utilization problem, which is around 70% in the average case. Therefore, the authors improve the insertion mechanism of R-trees to a more compact R-tree structure, with no penalty on performance during queries.

Motivated by the analogy between separating of R-tree node entries during the split procedure on the one hand and clustering of spatial objects on the other hand, Brakatsoulas et al. [3], implemented a novel splitting procedure that results in up to k nodes ($k \geq 2$ being a parameter), using the k-means clustering as a working example.

Static R-Trees

There are many applications that index only static data. For instance, insertions and deletions in census, cartographic and environmental databases are rare. For such applications, special attention should be paid to construct an optimal structure with regard to some tree characteristics, such as storage utilization maximization, minimization of overlap or coverage between tree nodes, or combinations of these. Methods for constructing static structures are well known in the literature as *packing* or *bulk loading* techniques.

The Packed R-Tree The first packing algorithm for R-trees was proposed by Roussopoulos and Leifker [20]. Let the data-set contain N rectangles and each page can store up to M rectangles. First, sort the rectangles in the data-set according to the x-coordinate of their centroid (equivalently, the x-coordinate of the lower-left corner can be used). Pack the rectangles of the data-set into $\lfloor N/M \rfloor$ consecutive groups of M rectangles (except the last group). Recursively pack the resulting groups into nodes. The recursion continues until a single root node is created.

The Hilbert Packed R-Tree The Hilbert Packed R-tree [12] resembles its dynamic counterpart, because it utilizes the concept of the Hilbert space-filling curve. The algorithm builds a static R-tree with nearly 100% storage utilization. The tree is constructed in a bottom-up manner. First, the rectangles are sorted using the Hilbert space filling curve of their centroids. Then, they are grouped recursively into nodes, like in the Packed R-tree.

The STR R-Tree STR (Sort-Tile-Recursive) is a bulk-loading algorithm for R-trees proposed by Leutenegger et al. [17]. Let N be a number of rectangles in two-dimensional space. The basic idea of the method is to tile the space by using VS vertical slices, so that each slice contains enough rectangles to create approximately $\sqrt{N/M}$, where M is the R-tree node capacity. Initially, the number of leaf nodes is $n_l = \lceil N/M \rceil$. First, the rectangles are sorted with respect to the x-coordinate of the centroids and VS slices are created. Each slice contains $VS \cdot M$ rectangles, which are consecutive in the sort order. In each slice, the objects are sorted once more, this time by the y-coordinate of the centroids and are packed into nodes (placing M objects in a node). The procedure continues recursively.

Time-Evolving R-Trees

There are many applications that manage data whose geometry changes over time. Examples include global change data (climate and land cover changes), transportation (traffic surveillance and intelligent transportation systems), social data (demographic and health), and multimedia data (movies). In such applications, object positions and/or extents change over time. A straightforward approach for indexing such data is to consider time as another dimension and use traditional R-trees. Nevertheless, this approach leads to extensive object overlapping (due to the long time intervals that have to be indexed) and, hence, deteriorated query performance. Thus, special R-trees have been developed for time-evolving applications.

The Segment R-tree attempts to resolve this problem by fragmenting long intervals into smaller, more manageable ones [15]. Two other approaches proposed are the *Overlapping* and the *Multiversion* R-trees. Consider a collection of spatial objects at two consecutive time instants t_1 and t_2; let one R-tree indexing the objects at t_1 and a second R-tree at t_2. If there are few changes between instants t_1 and t_2, then the corresponding R-trees will not be very different. The Overlapping R-trees take advantage of this observation by building a single structure for all consecutive R-trees that share common sub-trees. Multiversion R-trees use the notion of *partial persistence*. A data structure is called *persistent* if an update creates a new version of the data structure while previous versions are retained and can still be traversed [4]. *Partial* persistence implies that only the newest version of the structure can be modified. In the partial persistence approach, the sequence of R-trees can be visualized as the temporal evolution of the initial R-tree (the sequence of updates to the initial structure). The major advantage of partial persistence is that it uses space linear to the number of changes in the evolution of the objects while it provides access to any time instant with the same asymptotic efficiency of a dedicated R-tree for that time instant. Multiversion R-trees were introduced by Kumar et al. [16] for bi-temporal objects. Kollios et al. [14] applied them for spatio-temporal objects. Optimized approaches appear in [10,24].

A related application is that of indexing moving objects. Objects move with a known (typically linear) function of time, updating their position and movement function characteristics when necessary. One method proposed for indexing moving objects is the Time-Parametrized R-tree (TPR-tree) which indexes the current and anticipated future positions of objects [21]. A characteristic of TPR-trees is that they employ time parametrized MBRs that evolve over time as the object they contain move.

Query Optimization

Determining the best execution plan for a spatial query requires tools for estimating the number of data items that are retrieved by a query as well as its I/O and CPU cost. For R-tree indices, cost-based optimization exploits analytical models and formulas that predict the number of hits among the entries of the R-tree (called *selectivity*) and the cost of a query retrieval, measured in node accesses (or actual disk accesses, assuming existence of a buffering scheme).

The expected height h of an R-tree is $h = \log_f \frac{N}{M}$, where f is the average fan-out for parent nodes, M is the capacity of leaf nodes, and N is the number of data entries. The average cost C_W of a range query with respect to a query window $q = (q_1, \ldots, q_d)$, (for a d-dimensional R-tree), and provided that the sides $(s_j, 1, \ldots, s_j, d)$ of each node s_j are known is:

$$C_W(q) = \sum_j \left\{ \prod_{i=1}^{d} (s_{j,i} + q_i) \right\} . \tag{1}$$

Equation 1 allows the query optimizer to estimate the cost of a range query (measured in number of node accesses), if the MBR of each node s_j of the R-tree can be measured. The equation intuitively presents the relation between the sizes of the R-tree nodes and of the query window with the cost of a range query. Moreover, the influence of the node perimeters is revealed, explaining the efficiency of the R*-tree [6].

In order to avoid having to compute the R-tree node extends for estimating the cost of a query, the fractal dimension of a data-set can be used. The fractal dimension is a simple way to describe non-uniform data-sets, using just a single number. According to the model proposed in [5], the estimation of the number of disk accesses at level 1 (i. e., leaf level), denoted by $C_W(1, q)$, is given by:

$$C_W(1, q) = \frac{N}{f} \cdot \prod_{i=1}^{d} (s_{1,i} + q_i) \tag{2}$$

where $s_{1,i} = (f/N)^{1/fd}$, $\forall i = 1, \ldots, d$ and f is the average fan-out of the R-tree nodes.

R

A different approach is to use another property of the data-set, called *density surface* [26]. The density D of a set of N rectangles with average extent $s = (s_1, \ldots, s_d)$ is the average number of rectangles that contain a given point in d-dimensional space. Equivalently, D can be expressed as the ratio of the global data area over the work space area. Consider a unit workspace $[0,1)^d$ then the density $D(N,s)$ is given by the following formula:

$$Den(N,s) = \sum_N \prod_{i=1}^{d} s_i = N \cdot \prod_{i=1}^{d} s_i. \qquad (3)$$

Given the observations that: 1. The expected number of node accesses $C_W(q)$ for a query q is equal to the expected number of intersected nodes at each level; 2. The average number of intersected nodes is equal to the density D of the node rectangles inflated by q_i at each dimension; 3. The average number of nodes n_j at level j is $n_j = N/f^j$, where N is the cardinality of the data-set and f is the fan-out; 4. The density D_j of node rectangles at each level j can be expressed as a function of the density of the data-set, the following formula can be derived:

$$C_W(q) = \sum_{j=1}^{1+\log_f \frac{N}{f}} \left\{ \frac{N}{f^j} \cdot \prod_{i=1}^{d} \left(\left(Den_j \cdot \frac{f^j}{N} \right)^{1/d} + q_i \right) \right\}. \qquad (4)$$

To reach this formula, square node rectangles and uniform data distribution for both data and node rectangles are assumed. Further work has provided formulas that drop the uniformity assumption both for data and internal nodes.

Key Applications

Processing Spatial Queries

R-trees can be used for processing basic and complex spatial queries. Basic query types include range and topological queries (for identifying data that satisfy a query region, along with their topological relations with respect to this region), nearest-neighbor queries (for identifying data closest to a query point), and spatial join queries (for finding pairs of data that satisfy a condition, e. g., "determine all hotels near a given location"). Complex queries include categorical range queries (identifying groups of objects that intersect with the query), reverse and constrained nearest neighbor queries (identify the set of objects that have the query point as a nearest neighbor), multi-way spatial join queries (joining between multiple inputs), incremental distance-join (find a subset of the Cartesian product that satisfies an order based on distance), closest-pair queries (find the k pairs of objects with the smallest distance), and all-nearest neighbor queries (for every object in set A identify its nearest neighbor in set B). Efficient implementations of R-tree variants for the above queries can be found in [25].

Spatio-temporal Databases

Spatio-temporal database systems aim at combining the spatial and temporal characteristics of data. There are many applications that benefit from efficient processing of spatio-temporal queries such as: mobile communication systems, traffic control systems (e. g., air-traffic monitoring), geographical information systems (GIS), multimedia, and location-based services (LBS). The common basis of these applications is the requirement to handle both the space and time characteristics of the underlying data. These applications pose high requirements concerning the data and the operations that need to be supported, and therefore new techniques and tools are needed for increased processing efficiency. A large number of specialized indexes for spatio-temporal data management are based on R-trees. Efficient implementations of spatio-temporal R-tree variants are found in [9].

Multimedia Databases

Multimedia database management systems aim at the effective representation and the efficient retrieval of multimedia objects, such as text, images, audio, and video. The basic characteristics of multimedia objects that makes multimedia query processing challenging are the following. They are characterized by significant complexity due to their rich content, and hence are hard to be organized and managed. The notion of similarity between two multimedia objects is often difficult to express. Therefore, new algorithms are required to search multimedia databases based on content. R-trees have been successfully used to alleviate these problems.

Data Warehousing and Data Mining

Data warehouses are specialized databases that serve as repositories for multiple heterogeneous data sources, organized under a unified schema to facilitate decision making. On-Line Analytical Processing (OLAP) is an analysis technique performed in data warehouses, which is exploratory-driven. Data mining (or knowledge discovery in databases) is the process of extracting interesting information or patterns from data in large databases. Both data warehousing and data mining need to access large amounts of data. For this reason, the acceleration of data access is facilitated by indexes. Although specialized indexes have been developed for these new environments (e. g., bitmap indexes), R-tree-like indexes have also been successfully used.

Future Directions

The R-tree is one of the most influential Spatial Access Methods and has been adopted as the index of choice in many research works regarding spatial and multidimensional query processing. Taking into consideration the work performed so far, the R-tree stands for the spatial databases what the B-tree represents for alphanumeric data types. Considering the rich work performed on R-trees so far, it clearly covers almost all aspects concerning a database system: query processing, query optimization, cost models, parallelism, concurrency control, and recovery. This is the main reason why gradually database vendors adopted the R-tree for spatial data management purposes. Conclusively, the R-tree family is of great theoretical interest from several points of view. However, in view of new emerging technologies and applications, the R-tree will become a ubiquitous data structure. Taking into consideration the significant research performed on R-trees during the last 20 years, one could conclude that there is no more room for improvements or variations. However, current research in the area proves exactly the opposite. There are new and exciting application domains that require efficient representation and processing of objects that can be handled by R-tree-based access methods. Modern areas of increasing importance, such as P2P systems, data streams, and bio-informatics, are making use of R-trees as a powerful data management mechanism. It is sound to anticipate that in the future R-trees will be granted an exalted position in modern systems and applications.

Cross References

▶ Continuous Queries in Spatio-temporal Databases
▶ Indexing and Mining Time Series Data
▶ Indexing, High Dimensional
▶ Indexing, Hilbert R-tree, Spatial Indexing, Multimedia Indexing
▶ Indexing Schemes for Multi-dimensional Moving Objects
▶ Indexing Spatio-temporal Archives
▶ Indexing the Positions of Continuously Moving Objects
▶ Indexing, X-Tree
▶ Mobile Object Indexing
▶ Nearest Neighbor Queries in Network Databases
▶ Nearest Neighbor Query
▶ Pyramid Technique
▶ Quadtree and Octree
▶ Queries in Spatio-temporal Databases, Time Parameterized
▶ R*-tree
▶ Spatial Data, Indexing Techniques
▶ Trip Planning Queries in Road Network Databases

Recommended Reading

1. Ang, C.-H., Tan, T.C.: New linear node splitting algorithm for r-trees. In: Proc. of Symposium on Advances in Spatial Databases (SSD), Berlin, Germany, 15–18 July 1997, pp. 339–349
2. Beckmann, N., Kriegel, H., Schneider, R., Seeger, B.: The R*-tree: An efficient and robust access method for points and rectangles. In: Proc. of ACM Management of Data (SIGMOD), New Jersey, USA, 23–25 May 1990, pp. 220–231
3. Brakatsoulas, S., Pfoser, D., Theodoridis, Y: Revisiting r-tree construction principles. In: Proc. of the East European Conference on Advances in Databases and Information Systems, Bratislava, Slovakia, 8–11 Sept 2002, pp. 149–162
4. Driscoll, J.R., Sarnak, N., Sleator, D.D., Tarjan, R.E.: Making data structures persistent. J. Comp. Syst. Sci. **38**(1):86–124 (1989)
5. Faloutsos, C., Kamel, I.: Beyond uniformity and independence: analysis of r-trees using the concept of fractal dimension. In: Proc. of ACM Symposium on Principles of Database Systems (PODS), Minnesota, USA, 24–26 May 1994, pp. 4–13
6. Faloutsos, C., Sellis, T., Roussopoulos, N.: Analysis of object oriented spatial access methods. SIGMOD Record **16**(3):426–439 (1987)
7. Garcia, Y.J., Lopez, M.A., Leutenegger, S.T.: On optimal node splitting for r-trees. In: Proc. of Very Large Data Bases (VLDB), New York, USA, 24–27 Aug 1998, pp. 334–344
8. Guttman, A.: R-trees: A dynamic index structure for spatial searching. In: Proc. of ACM Management of Data (SIGMOD), Massachusetts, USA, 18–21 June 1984, pp. 47–57
9. Hadjieleftheriou, M., Hoel, E., Tsotras, V.J.: Sail: A library for efficient application integration of spatial indices. In: Proc. of Scientific and Statistical Database Management (SSDBM), Santorini Island, Greece, 21–23 2004, pp. 135–138
10. Hadjieleftheriou, M., Kollios, G., Tsotras, V.J., Gunopulos, D.: Efficient indexing of spatiotemporal objects. In: Proc. of Extending Database Technology (EDBT), Prague, 24–28 Mar 2002, pp. 251–268
11. Huang, P.W., Lin, P.L., Lin, H.Y.: Optimizing storage utilization in r-tree dynamic index structure for spatial databases. J. Syst. Softw. **55**(3):291–299 (2001)
12. Kamel, I., Faloutsos, C.: On packing r-trees. In: Proc. of Conference on Information and Knowledge Management (CIKM), Washington DC, USA, 1–5 Nov 1993, pp. 490–499
13. Kamel, I., Faloutsos, C.: Hilbert r-tree: An improved r-tree using fractals. In: Proc. of Very Large Data Bases (VLDB), Santiago de Chile, Chile, 12–15 Sept 1994, pp. 500–509
14. Kollios, G., Tsotras, V.J., Gunopulos, D., Delis, A., Hadjieleftheriou, M.: Indexing animated objects using spatiotemporal access methods. IEEE Transactions Knowl. Data Eng. (TKDE) **13**(5):758–777 (2001)
15. Kolovson, C., Stonebraker, M.: Segment Indexes: Dynamic indexing techniques for multi-dimensional interval data. In: Proc. of ACM Management of Data (SIGMOD),Colorado, USA, 29–31 May 1991, pp. 138–147
16. Kumar, A., Tsotras, V.J., Faloutsos, C.: Designing access methods for bitemporal databases. IEEE Transactions Knowl. Data Eng. (TKDE) **10**(1):1–20 (1998)

17. Leutenegger, S.T., Edgington, J.M., Lopez, M.A.: Str: A simple and efficient algorithm for r-tree packing. In: Proc. of International Conference on Data Engineering (ICDE), Birmingham, UK, 7–11 Apr 1997, pp. 497–506
18. Manolopoulos, Y., Nanopoulos, A., Papadopoulos, A.N., Theodoridis, Y.: Rtrees: Theory and Applications. Springer, Germany (2005)
19. Nanopoulos, A., Vassilakopoulos, M., Manolopoulos, Y.: Performance evaluation of lazy deletion methods in r-trees. GeoInformatica **7**(4):337–354 (2003)
20. Roussopoulos, N., Leifker, D.: Direct spatial search on pictorial databases using packed r-trees. SIGMOD Record **14**(4):17–31 (1985)
21. Saltenis, S., Jensen, C.S., Leutenegger, S.T., Lopez, M.A.: Indexing the positions of continuously moving objects. SIGMOD Record **29**(2):331–342 (2000)
22. Schreck, T., Chen, Z.: Branch grafting method for r-tree implementation. J. Syst. Softw. **53**(1):83–93 (2000)
23. Sellis, T., Roussopoulos, N., Faloutsos, C: The r+-tree: A dynamic index for multi-dimensional objects. In: Proc. of Very Large Data Bases (VLDB), pp. 507–518 (1987)
24. Tao, Y., Papadias, D.: MV3R-Tree: A spatio-temporal access method for timestamp and interval queries. In: Proc. of Very Large Data Bases (VLDB), pp. 431–440 (2001)
25. Theodoridis, Y.: The R-tree-Portal. (2003)
26. Theodoridis, Y., Sellis, T.: A model for the prediction of r-tree performance. In: Proc. of ACM Symposium on Principles of Database Systems (PODS), pp. 161–171 (1996)

Rubber Sheeting

► Positional Accuracy Improvement (PAI)

Rubber-Sheeting

► Conflation of Features

Rules-Based Processing in GIS

MICHAEL SANDERSON, PETER WOODSFORD
Laser-scan, Cambridge, UK

Synonyms

Digital mapping

Definition

This paper describes how Laser-Scan developed products to meet the GIS market requirements during the period 1980–2007. From an early background in the intelligent vectorization of maps, Laser-Scan's product philosophy has been to provide solutions that maximize the degree of automation in spatial data handling. The ever-increasing size of spatial databases mandates this approach, as well as the risk of human error associated with manual intervention. However, it is fully recognized that human interaction is required to handle cases that cannot be resolved. Much of the automation proceeds from the application of algorithms. Topology management plays a significant role. Successive generations of product have been characterized by increasing reliance on standards and mainstream database technology and by more formalized implementations of rules-based processing. Typical applications include data quality assessment, validation and maintenance services, multi-product information management and generalization. Up to date information is available via [5].

Historical Background

Laser-Scan's GIS products have evolved across three generations, characterized by increasing alignment with mainstream Information Technology and hence by the exploitation of the dramatic advances in hardware, software and IT infrastructure that have taken place since the company was formed in 1969.

The first generation products were essentially digital mapping systems, and were based on the company's capabilities in automated data capture from maps. Initially this used special purpose laser-scanning technology, to follow lines and edges and to recognize symbols on film versions of maps. Later the algorithms were adapted to extract intelligent, quality assured vector information from raster data [4] in the VTRAK product. VTRAK was complemented by a powerful vector editing system that formed LAMPS (Laser-Scan Automated Map Production System). Vector data were held in files, in a proprietary format with a publicly accessible API. Raster edit and burn-in capabilities were also supported. LAMPS was integrated with powerful map finishing systems such as Mercator (now part of STAR INFORMATIC). A number of cartographic and charting systems remain in production use with considerably more than 20 years of service. Scripts and macros were used extensively to control sequences of automated operations on the data and to enforce business rules, but these were in not formalized in any meaningful sense.

The advent of the object-oriented (O-O) paradigm provided a more formalized platform for rules-based processing and handling of structured information. It embodied the thinking behind the mantra:

Model = **D**ata + **S**tructure + **A**lgorithm [13]

Starting in the late 1980s Laser-Scan developed the Gothic geospatial object-oriented processing environment. This approach was similar to Intergraph's TIGRIS and to Smallworld GIS.

Gothic represents real world objects by object models in classes, according to a schema. Instances of objects possess identity, geometry, attributes and behaviors. In addition relationships between objects, including topology, can be represented and complex objects (objects made up of a group of objects) constructed and maintained. A comprehensive set of geoprocessing operations is provided, together with support for object lifecycles. A proprietary user language (LULL) was developed as both the scripting language and the methods language in the Gothic object database. The use of methods led to the formalization of business logic, constraints and quality rules and their enforcement centrally across all applications using the rules-processing environment provided by the Gothic database.

Two concepts of the object lifecycle approach are worthy of further discussion, since they point the way to the future architecture. They are encapsulation and polymorphism. Encapsulation standardizes and simplifies access to data objects and protects them. The data stored inside the objects can be manipulated safely by recourse to the methods, which the object class provides or inherits. For query, complex derived attributes (e. g. area) may be computed from object data (e. g. geometry), simplifying maintenance. Typically the method behavior is of sufficient complexity that attempting to reproduce it in application logic risks data corruption or performance degradation. Most importantly, encapsulation promotes thin clients. Polymorphism is the ability to use the same syntax for objects of different types. A good example is a validation method. When editing data, the application may wish to validate the feature being edited before committing it to the database. Depending upon whether the feature is of type motorway (inherits the road validation method) or of type contour line (inherits the relief validation method) it will invoke a completely different behavior. Provided the validate method is polymorphic the application need know nothing about the internals of how validation is accomplished. Again thin client delivery is appropriate because all the geoprocessing has been executed in the database and can be shared and accessed by all clients. Polymorphism and encapsulation are well suited therefore to the current web based IT architecture.

Two aspects of Gothic have become less viable. The use of a proprietary language was not sustainable in the face of the widespread adoption of O-O programming languages such as Java. LULL has given way to Java, firstly as the scripting language and more recently as the methods language. Thus a modern Gothic application such as the Air Information Management System is written at the application level entirely in Java. Equally the use of a proprietary database became unacceptable as users sought the integration of their geospatial information holdings with all other business information. Enterprise databases are delivered with features such as security and scalability as standard and with a large pool of supporting specialists. Rather than replicating these features, it is better to seek the GIS value add in managing the spatial data.

The database storage issue has resulted in a new Laser-Scan architecture. The re-architecture is based on a mainstream database (typically Oracle, with the advent of Oracle Spatial). However it is something more fundamental than simply that. It is more fully understood as a requirement for the deployment of powerful geo-processing capabilities as components in an open, standards-based architecture. This provides the rationale for the Radius suite products, launched in 2002.

Scientific Fundamentals

The O-O techniques combined with persistent topology provide the basis for managing spatial data. This section opens with a description of key functionality in the Radius product suite. As described above the Radius products are a subset of the Gothic environment.

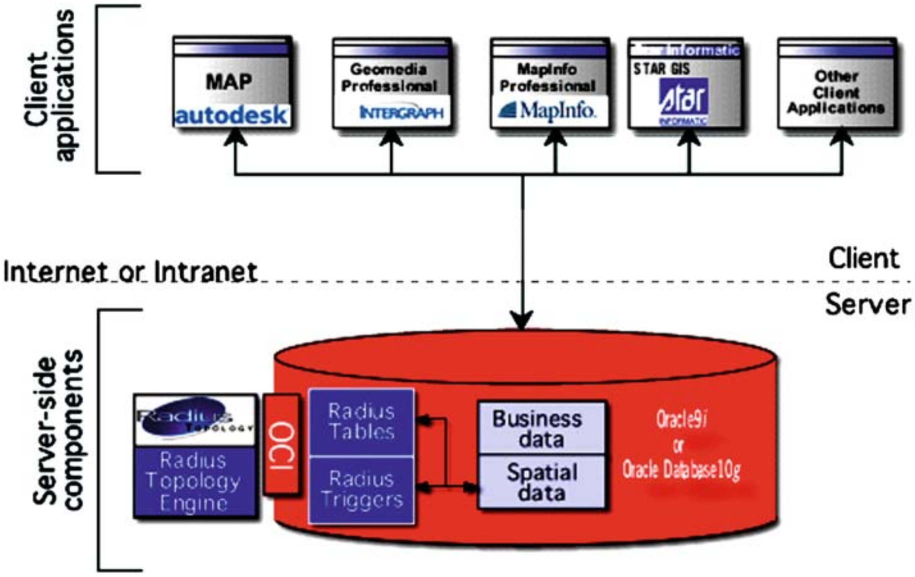
Radius Topology

The Laser-Scan topology engine implements the 2D components of the ISO19107 topology model [3]. Radius Topology delivers powerful data cleaning, connectivity and analysis functionality to spatial data held in Oracle9*i* or Oracle Database10*g*, through the SDO_Geometry interface [12]. The design (see Fig. 1) is based on avoiding any impact on other GIS applications reading/writing the standard spatial data types in Oracle according to whatever long transaction model has been determined. There is a very fine level of control over topological relationships, which can be defined and prioritized on an individual class-class basis, and which are maintained 'on-the-fly' across all processes. In addition, support is provided for directly editing topological primitives.

In the real world GIS data has been collected with a height value (sometimes known as 2.5D data). Radius Topology therefore implements topological structuring with a height (or z value) for each vertex, using vertical snapping tolerances and maintaining relationships such as 'over/under'.

Over time support for specific vertical market applications has emerged. Thus pre-configured tolerances and priorities have been defined to support utility connectivity, network conflation, land and property gazetteer creation and maintenance. These are termed Wrappers and provide a simple Java user interface and a PL/SQL API.

R



Rules-Based Processing in GIS, Figure 1 The Radius Topology Architecture

Radius Studio

Topological structuring is an automated process. Often what is required is a meaningful assessment or statement of the data quality position – a window on the data. This assessment can be used to define remedial programmes and to assess the fitness for purpose of the data set in particular decision-making scenarios. Radius Studio is such an assessment tool. It makes use of the 3-tier architecture that is emerging as part of the web services (component) paradigm. This means it has to operate within a standards-based framework. It is based on the following de facto and de jure standards:

- Component model – J2EE/EJB;
- Service model – ISO 19119;
- Geometry model – ISO 19107;
- Domain modelling – W3C OWL;
- Rules language – W3C SWRL.

Radius Studio consists of a rules browser/editor, conformance checker, a spatial data processing and reconciliation tool and a task sequencing tool. It has been designed on knowledge management principles [8] to allow the domain expert to define the business rules. It offers an approach to semantic interoperability, as well as measurable spatial data quality control. A common language interface is used in preference to a programmatic interface (see Fig. 2).

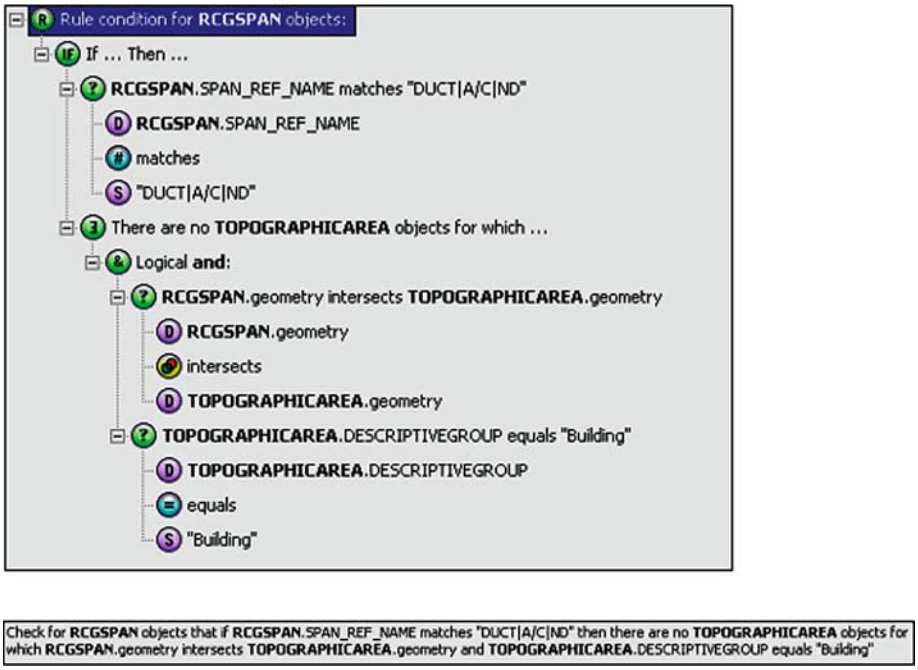
Gothic

Laser-Scan offers a range of technologies based on its Gothic O-O database, from an application development environment to tools for web mapping. The Gothic Database Server plays a central role, providing access management for a versioned, object-oriented spatial database (OODB). Spatial objects, in 2D or 3D, are members of object classes defined in a schema. Methods are defined for object classes – these are rules or behaviors written in the internal language or in Java. Methods have access to the full range of geospatial processing capabilities. Encapsulation is supported within the object-lifecycle framework. Combining data and functionality in this way ensures that Gothic datasets are self-validating. Rich data models with complex relationships, rules and constraints can be implemented and maintained across a wide set of applications.

Gothic databases are seamless and can hold very large continuous datasets with many millions of objects. The topology engine is the same set of libraries that are described above for Radius Topology only stored and accessed in a different manner. Objects can be stored in geographical coordinates, with whole earth support (crossing polar regions and dateline), or in any coordinate projection. They can be stored in one coordinate system, structured and queried in another, and displayed in another. A full set of projection and map transformations are supported. States of an object can be date-stamped to support temporal evolution. In addition to vector objects, Gothic databases can hold and manage raster and image objects.

The Gothic Developer Application Programming Interface (API) provides a bridge between Laser-Scan's O-O spatial database capabilities and the requirements of application builders. Using Developer, applications that access the Gothic O-O database can be written or customized.

LAMPS2 is an advanced production application for spatial data sets, maps, charts, plans or any other geographical data product. It separates the tasks of compilation and update from those of presentation and product generation,



Rules-Based Processing in GIS, Figure 2 A fragment of a rule definition in Radius Studio

and supports data re-engineering and structure-building. Raster/vector integration includes raster edit and burn-in capabilities.

Gothic JADE offers a completely extensible Java-based desktop application providing spatial and mapping capabilities. It provides access to the Gothic spatial toolkit and object database from a framework application into which standard or user-written modules can be loaded.

Key Applications

A subset of applications is described, with an emphasis on the role of rules-based processing and starting with the Radius suite.

Topology and its Uses

Topology is defined elsewhere. Using indexing Laser-Scan utilizes the SDO Geometry construct in Oracle to calculate and persist topology on the server side. The main reason for this is to provide interoperability with business applications. An alternative view is described in [2]. The uses of topology in GIS include the analysis and maintenance of networks, routing applications, efficient maintenance of shared boundaries in for example land parcel applications and the speeding up of spatial database queries [15].

Positional Accuracy Improvement (PAI)

Positional accuracy is one of the data quality elements covered by ISO 19113. The advent and widespread use of high precision GPS has meant that positional inaccuracies in historical survey data have been exposed and become a problem. The vast majority of the spatial data in use were collected before GPS was ubiquitous. In Europe alone the value of these data were €36bn in 1999 [1]. Topology can be used effectively to solve part of this problem by establishing the relationships in the data before applying shift vector transformations to improve positional accuracy. This allows the pre-GPS data to be re-used. A Radius Topology PAI Wrapper tracks and maintains the relationships between asset data and base data, keeps asset data connected during shifting and audits relationships before and after shift.

Data Quality Certification and Maintenance

There is a fundamental issue over fitness for purpose in the use of spatial data in the decision-making process. In every day use or in natural disaster response management the value of the data are undermined if not updated and managed over time. In a similar manner to the GPS issue discussed above, previous generations of software tools have seen the business rules of an organization locked into application code, in graphical attribution. In addition there has been little rigour in spatial data administration. To create true knowledge economies business rules need to be abstracted and made available throughout the enterprise by the domain experts. Most organizations are asset intensive. Assets have spatial attributes. These spatial data need to conform to certain business rules. Also the cost of collection and subsequent maintenance of these data needs to be measured against quality measures. Examples of such measures may be the number of dry holes that are dug to locate the company's plant, or the number of customer refunds paid as a result of outages not identified in network maintenance or repair programmes. A rules-based approach in the latter example (see the rule in Fig. 2) would establish the connectivity between the property and the asset as a data management problem. The results of these analysis will in the future need to be a part of the key performance indicators against which an organization measures itself.

Multi-product Information Management Systems

Concentration of business logic on the server side, together with rich data modelling capabilities, underpins the implementation of multi-product information management systems. The key characteristics are once only database maintenance with assured data quality, multi-product capabilities and consistent and efficient product provision. An example is the Aeronautical Production System (APS) [6]. This produces a quality assured set of Flight Information products including cartographic products (en route charts), data products (DAFIF, DVOF) and textual reports automatically from a single database.

In APS a formally defined data model (DAFIF) drives both the data model schema and the lifecycle and validation rules (see Fig. 3). All data entering the system is validated against the DAFIF rules. Product Data Rules extract information from the database by combinations of operations such as selection of a temporal snapshot of a localized geographic region, transformation of specific features and attributes and topological structuring of geometric features. Product Cartographic Rules define automatic and intelligent feature portrayal. An example is shown in Fig. 4.

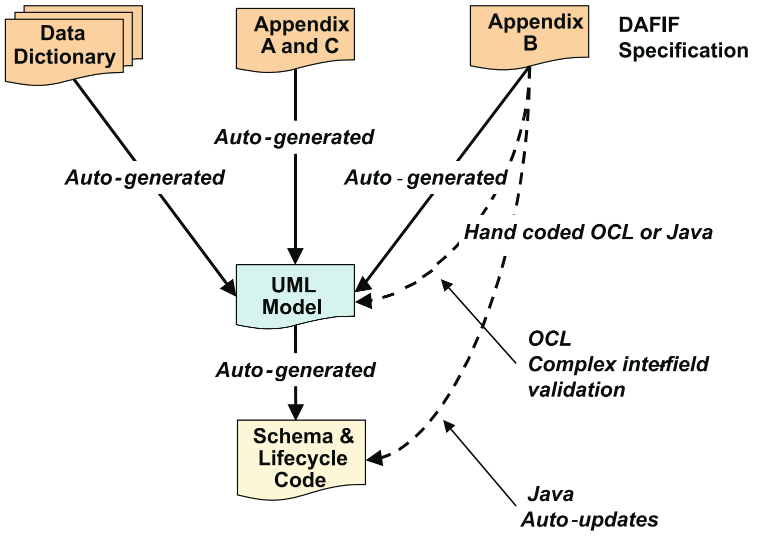
Data Model Re-engineering

Data providers sometimes take a quantum leap and seek to re-purpose their data by adopting a richer and more capable model. A move from spaghetti data to structured real world objects with persistent identifiers is a prime example of such re-purposing. Rules-based processing is key to achieving such re-engineering in a manageable timescale and with assured quality. A major example is the Ordnance Survey Great Britain (OSGB) project to re-engineer Land-Line data (220,000+ sheets of points and lines) to MasterMap® data (continuous, 0.5 billion richly attributed polygon objects with public identifiers (TOIDs™)). This was achieved by an automated flowline using Gothic in one year, with less than 1% manual editing [11].

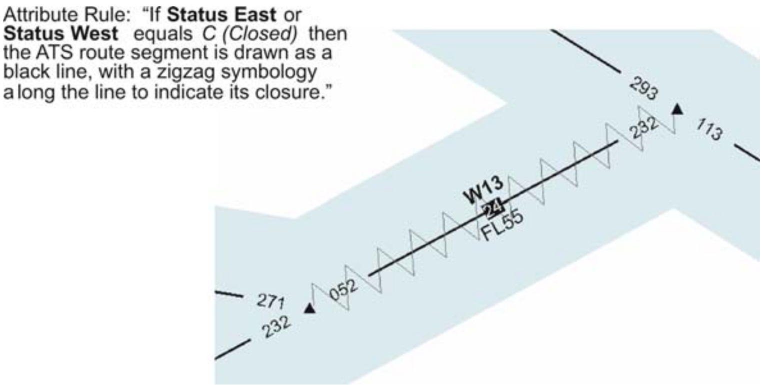
Generalization

Laser-Scan's *Clarity* is a cartographic and data generalization product built on the Gothic intelligent geo-spatial functionality. It automatically reduces data resolution (and hence volume) whilst retaining essential information content by applying generalization rules [9]. For cartographic purposes Clarity can also apply rules to enhance graphical clarity and readability using active, goal-seeking, 'self aware' agents. This can result in the displacement of features, in addition to other generalization processes such as selection and amalgamation.

Processing rules and constraints that must be maintained are defined by a simple user interface and encoded in



Rules-Based Processing in GIS, Figure 3
Aeronautical Production System – model driven definition of schema and rules base



Rules-Based Processing in GIS, Figure 4 A cartographic rule on an aeronautical chart detail

XML. Generalization is performed automatically, but with exception handling. Instances that cannot be processed automatically or which do not conform to product rules are flagged for subsequent operator action.

Data generalization for example from 1:10K to 1:50K in the German Länder project [14] is typically achieved with complete automation. Cartographic generalization still involves some operator action but time savings in a production environment of an order of magnitude have been achieved, e. g. IGN France Carto2001 and New Base Map projects [7].

Future Directions

Much work has been carried out on spatial data models, by the OGC and others, but there are several areas where future progress is essential. There are five areas that are of significance: Progress towards 3D capabilities; Master Data Management; spatial as mainstream data, ontologies and semantics and Service Oriented Architecture (SOA).

3D

Progress towards 3D capabilities is rapid, spurred on by the advent of Google Earth. From our perspective the key issues are the extension of the topology engine, which currently handles 3D data but restricted to 2.5D topology, to a full 3D capability, in line with ISO19107.

Master Data Management

There is little real progress towards interoperability. Significant work has been undertaken in mainstream supply chain processes to establish the concept of Master Data. Central control is required to establish meta referencing to ensure that all participants in the supply chain have a common understanding of transactional events. It is essential to have high quality base reference mapping to provide the gold standard for data and against which to align any incoming or third party data.

Integration to Business Intelligence (BI)

Further work on the smooth integration of spatial models with other domains, particularly BI is essential if the spatial community is to be able to demonstrate its capability in delivering benefits across the organization. There is no reason why C-level executives current predilection for managing performance through dashboards, should not include spatial data measures. Given the expenditure on spatial data [1] there must be measurement. A Topological Quality Assessment Service is being implemented in the OGC OWS-4 testbed as an initial step to defining interface standards for quality assessment services [10].

Ontologies and Semantics

The model driven approach will become more powerful and widespread as better tools for managing ontologies emerge. Radius Studio can currently use an ontology defined in OWL but few of these are yet available. These developments will also result in richer discovery metadata and progress towards the Semantic Web in the geospatial domain.

Web Services Architecture

Radius Studio can be used over the web, as a service. The approach by Laser-Scan to the component-based web architecture has led towards a different business model for the company. This new approach is based on partnering. Some 37 years after Laser-Scan was founded this new business model has led to the Radius brand assuming greater importance than the company name, which was changed at the end of 2006 to 1Spatial.

Cross References

► Computing Fitness of Use of Geospatial Datasets
► Conflation of Geospatial Data

- ▶ Geospatial Semantic Integration
- ▶ Geospatial Semantic Web, Interoperability
- ▶ Map Generalization
- ▶ OGC's Open Standards for Geospatial Interoperability
- ▶ Ontology-Based Geospatial Data Integration
- ▶ Oracle Spatial, Geometries
- ▶ Positional Accuracy Improvement (PAI)

Recommended Reading

1. Commercial Exploitation of Europe's Public Sector Information. PIRA International, Leatherhead, UK (2000)
2. Hoel, E., Menon, S., Moorhouse, S.: Building a Robust Implementation of Topology. In Advances in Spatial and Temporal Databases. Proceedings of the 8th International Symposium on Spatial and Temporal Databases. SSTD 2003. Santorini, Greece (2003)
3. ISO 19107:2003. Geographic Information – Spatial schema. Available via http://www.isotc211.org/
4. Jackson, M.J., Woodsford, P.A.: GIS data capture hardware and software. In: Geographical Information Systems, pp. 239–249. Longman Scientific and Technical, Harlow, England and Wiley, New York (1991)
5. Laser-Scan. www.laser-scan.com and from December 2006, www.1spatial.com
6. Laser-Scan. The Aeronautical Production System (APS) (2003). http://www.laser-scan.com/applications/aps.htm
6. Lecordix, F., Trévisan, J., Le Gallic J.M., Gondol, L.: Clarity Experimentations for cartographic generalisation in production. 9th ICA Workshop of the ICA Commission on Generalisation and Multiple Representation, 25 June 2006 – Portland, USA. Available at http://ica.ign.fr/
7. Loshin: Enterprise Knowledge Management (The Data Quality Approach). Morgan Kaufmann, San Francisco (2001). ISBN 0-12-455840-2
8. Neuffer, D., Hopewell, T., Woodsford, P.: Integration of Agent-based Generalisation with Mainstream Technologies and other System Components. ICA Workshop on Generalisation and Multiple Representation 20–21 August 2004 Leicester, UK Also available at http://www.laser-scan.com/papers/Neuffer-v2-ICAWorkshop.pdf
9. OGC, Open Geospatial Consortium, Open Web Services, Phase 4 (2006), Topology Quality Assessment Service in Geoprocessing Workflow thread at http://www.opengeospatial.org/projects/initiatives/ows-4
10. Ordnance Survey Great Britain (GB). Meeting the Master Challenge. 18 Sept 2001. Available at: http://www.ordnancesurvey.co.uk/oswebsite/media/news/2001/sept/masterchallenge.html
11. Ravikanth, K.V., Godfrind, A., Beinat, E.: Pro Oracle Spatial: From Professional to Expert. Apress, Berkley (2005). ISBN 1590593839
12. Ross, D.T., Rodriguez, J.E.: Theoretical foundations for the computer aided design system, pp. 305–322. Proceedings of AFIPS 23, SJCC, Detroit (1963)
13. Urbanke, S.: AdV – Projekt "ATKIS – Generalisierung" 53. Deutscher Kartographentag, Rostock, 22 Sep 2005 (in German)
14. Watson P.: Topology and ORDBMS Technology, a White Paper, 2002. Available at http://www.laser-scan.com/pdf/topo_ordbms.pdf

Rural Hydrologic Decision Support

DAMON GRABOW
Regional Weather Information Center,
University of North Dakota, Grand Forks, ND, USA

Synonyms

Decision support; Rural hydrology; Soil and water assessment tool "SWAT"

Definition

Rural hydrologic decision support, in a general sense, is a tool or data set that supports decision makers with an emphasis on hydrology and its effects on other environmental factors. A hydrologic model such as the Soil and Water Assessment Tool (SWAT) could be referred to as a rural hydrologic decision support tool.

Historical Background

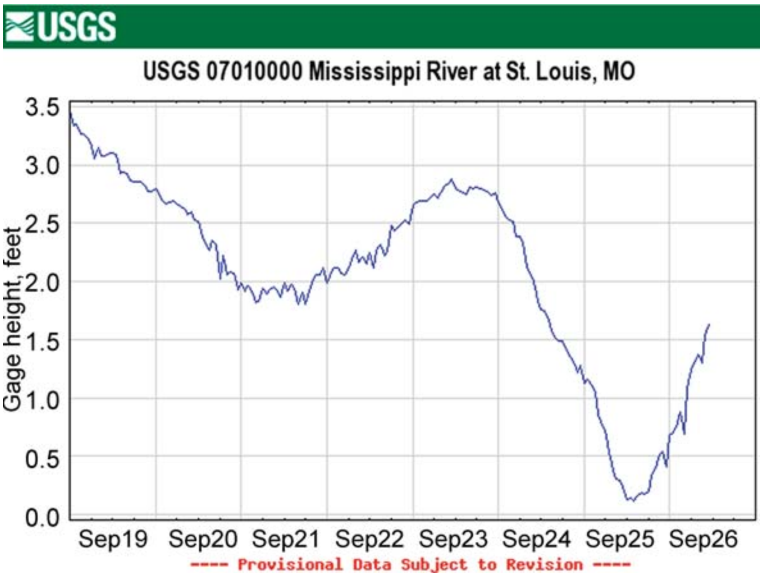
The typical data used for hydrologic decision support were streamflow datasets or possibly groundwater depth. For example, the United States Geological Survey (USGS) maintains a large number of stream gauges around the US. Figure 1 shows an example of stream gauge data output. Gauges, such as the one on the Mississippi River at St. Louis, MO, are placed along major river systems.

Scientific Fundamentals

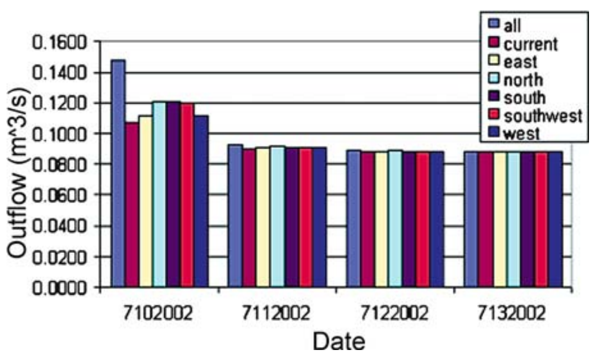
Rural Hydrologic Decision Support Example

One example of a rural hydrologic decision support is the growth of a small rural community during a climatic trend of heavy rainfalls. The community is looking to expand in the area with the least environmental impact. The example site is a rural community of approximately 1000 people that covers a land area of 100 hectares. The community is surrounded by farm fields allowing for ample growth provided the sale of the land is approved. There is a small coulee that runs through town with a late summer discharge of 0.1 $m^3 s^{-1}$. The only impervious surfaces are the rooftops and one paved road, all of the other roads and parking lots are gravel. Given the lack of impervious surfaces and size of the community is classified as low-density housing even though the lots are in 0.101 to 0.405 hectares in size.

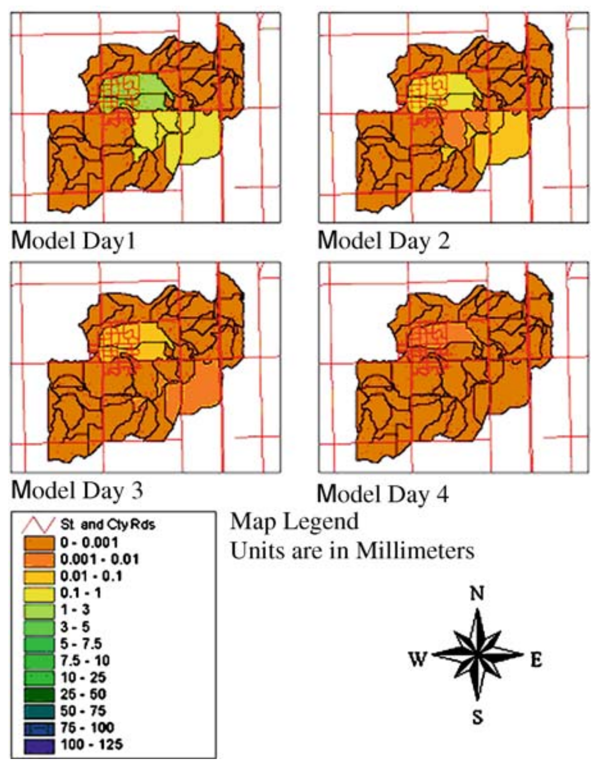
Topographically, the area is flat. A west to east slope is apparent as the western edge peaks at 267 m while the lower eastern edge shows an elevation of 259 m but has an average slope of 1.0–1.5 m per km. Due to the low slope,



Rural Hydrologic Decision Support, Figure 1 An example of a stream gauge data set. Data is from the USGS, http://waterdata.usgs.gov/mo/nwis/, gauge 70100000 along the Mississippi River at St. Louis, Missouri



Rural Hydrologic Decision Support, Figure 2 Total stream outflow from each urban scenario for 19.05 mm precipitation event



Rural Hydrologic Decision Support, Figure 3 Shown are spatial surface runoff results using the 19.05 mm precipitation event with the current urban area overlaid with current state and county roads

a few non-natural features are apparent in the digital elevation model; Railroad tracks and roadbeds are distinctly visible along with the interstate overpass.

For this example, 35 separate scenarios were established, 7 land use plans and 5 precipitations events. Along with the original land use, six possible urban land use areas were created for this study; south, north, east, southwest, west, and a combination of all of these. The precipitation events were of sizes 2.5 mm, 6.6 mm, 19.05 mm, and 44.45 mm, and 207 mm.

The results from this example experiment demonstrate the use a rural hydrologic decision support tool. Total stream outflow (Fig. 2) and estimated potential surface runoff (Fig. 3) were used to demonstrate the varying hydrologic impacts of land use change around the community.

By using the Soil and Water Assessment Tool (SWAT), one example of a hydrologic decision support tool, the community decision makers are able to visualize the areas of concern based on a wide variety of estimated variables. However, decision tools like SWAT need a compilation of large environmental data sets that include weather, hydrologic data, soils, land use and land cover, and elevation data.

A large chunk of these data can be obtained over the Internet for free depending on the users scaling needs.

Data

There are many pieces of data to gather in utilizing a decision support tool's full potential. For this example the following set of environmental data was gathered to initiate the support tool.

Growth Plan

If the user is a rural community it is highly beneficial to build a series of growth plans for comparison within the rural hydrologic decision support tool. Using a rural hydrologic decision support tool can estimate the environmental impacts of any land use changes or land use practices.

Meteorological Data

Meteorological data has a variety of variables. When dealing with hydrology precipitation is most common. Rain gauge data is the most widely used precipitation data type. Estimated precipitation from radar data is becoming available for the use within a GIS and provides for higher accuracy in hydrologic tools. However, other meteorological variables such as wind speed, temperature, cloud cover, dew point, and relative humidity can have an effect on the hydrologic changes over the course of time.

Hydrologic Data

Estimates of the existing surface water characteristics can be used to initialize a environmental model or equations. These estimates can consist of a stream flow rate, pond measurements, and even ground water levels. These data sets can be found on the web or estimated in the field with the use of some simple tools.

Soils

The types of soil can make a difference in the hydrologic response. Each soil type has its own hydrologic characteristics such as water capacity or infiltration rate, which are dependant upon the amount of clay, silt, or sand in the soil.

Land Use / Land Cover

The land use/land cover has an impact on the surface runoff, water quality, soil moisture, and even ground water accumulation. The land use can change the amount of impervious surface and have a direct impact to the hydrologic response. Vegetation types and patterns can alter the hydrologic response of a watershed as well. Any changes in land use and land cover should be addressed in the decision support data.

Elevation

Elevation data, typically found in grid form, is used to delineate watersheds and their sub-basins, locating outlets, and the hydrologic routing and flow patterns within the watersheds. It influences the speed at which the surface water will travel downstream and the erosion tendencies of the soil.

Key Applications

Rural hydrologic decision support can be used by any type of decision maker needing to make a decision relevant to hydrology.

Regional Planning/Rural Government

A large number of decision makers hold government offices and a large number of those offices represent rural areas. A hydrologic decision support tool/data set can be a benefit to a rural area and its growth and development plan. For example, by using the growth plan in a hydrologic decision support tool areas within that plan that produce the most hydrologic impact may be avoided. This may include topics such as land use management or flood retardants. Cau et al. (2003) used SWAT to estimate the water budget for the Sardinia region in Italy. Data for the project were a coarse soils map (1:250,000) and the CORINE Land Cover map (1:100,000), a 44-class vegetation and land use classification map. Regional rain gauge data and stream flow data were available dating back to 1922. The goal of the project was to construct a decision support tool that enables decision makers to use the water cycle as a variable in land use planning. Rainfall, evapotranspiration, and flow rate were the focus variables for the tool. By using the SWAT system for planning water management policies at a regional scale, decision makers can decide when, where and how water will be distributed. Cau et al. (2003) found that the model can play a leading role in estimating the water budget when the human utilization needs to be considered. A good description of the need for a rural hydrologic decision support system comes from a 2004 article written by Schär et al. This article explains the demise of the Aral Sea in central Asia that started in the early 1960's. Due to the agricultural demands in the Aral Sea watershed the rivers that replenished the sea from its evaporation were not carrying enough water to maintain the natural water levels. High water-demanding crops like cotton were the basis of the agricultural economy in the region. However, International governing bodies were able to change the agricultural focus to less water-demanding crops such as fruits, vegetables, and livestock reducing the use of irrigation. In order to maintain the water levels in the region meteorological models and hydrologic decision support tools were implemented to curb water usage and better manage the resources.

Hydrology

Melesse et al. (2003) researched the effects of land use and land cover change on runoff using the United States Department of Agriculture, Natural Resources Conservation Service Curve Number (USDA-NRCS-CN). The analysis was performed on four individual years, 1984, 1990, 1995, and 2000. Land cover and land use data were taken from the Landsat Thematic Mapper and Enhanced Thematic Mapper Plus of the Etonia, Econlockhatchee, and S-65A sub-basins in Florida. Elevation data were a 30-meter pixel grid obtained from the USGS. The chosen storm events, recorded by rain gauges, produced volumes of no less than 12.75 mm. In comparing the alterations in land-cover over the study years, the predicted runoff depth was shown to increase. The increased impervious surface area in 1995 and 2000 increased the peak flow, and reduces the time to peak and time to recession, compared to 1984 and 1990, respectively (Melesse et al. 2003).

Chemistry

Water quality is an important aspect of rural hydrologic decision support. With the use of GIS and spatial data models water quality can be modeled along with changes in land use or cover, surface water runoff chemistry, or even precipitation and snow melt. Rural hydrologic decision support can aid in water quality estimates and calculations. Chaplot et al. (2004) used a hydrologic decision support tool to investigate nitrogen applications and variations in agriculture practices. The nitrogen was varied by percentages of +20, +40, –20, –40, and –60 and the land use scenarios varied from pastures, corn-soybean rotations, and continuous winter wheat practices. Chaplot et al. found quantitative predictions of the impact of farming practices on annual and seasonal flow, NO_3-N, and sediment discharges and also explain that this technology should assist scientists, policy makers, and farmers in making decisions for management of watersheds.

Geology

In the same sense of a support tool or data set for surface hydrology, a support tool for hydrogeology can be used. Spatially based ground water models help supply information to decision makers regarding the effects on ground water, this being subsurface flow or water quality. Gutzler and Nims (2005) give an excellent example of one use of a hydrologic decision support tool. They discuss the groundwater depletion in Albuquerque, NM. Although Gutzler and Nims don't apply a hydrologic decision support tool they describe a groundwater issue that decision makers need more information about and is an opportunity to utilize a rural decision support tool.

Sun and Cornish (2005) researched the Liverpool Plains in Australia. Using a rural hydrologic decision support tool Sun and Cornish found that the groundwater recharge might not be as high as originally found in previous studies. The spatial characteristics over the entire catchment were included within the study (soil types, land cover, soil moistures, etc.). Sun and Cornish found that not only land clearing and crop practices contributed to the groundwater recharge but that climate variances did as well.

Eckhardt et al. (2003), also using SWAT, researched climate change impacts on groundwater recharge in central Europe. This research looked at the increasing amount of CO_2 in relationship with plant physiology and its affect on the groundwater recharge and change in streamflow. However, Eckhardt et al. also explains the complexity of using a hydrological decision support tool in that slight changes in the atmospheric boundary conditions can have a significant impact on the findings.

Ecology

Plants and animals are affected by decisions made by humans. Thus a rural hydrologic decision support tool can provide an estimate impact to change in habitat or conduct vulnerability assessments. Whittaker (2005) compared two remediation policies for salmon habitat within the Columbia River basin in Washington and Oregon. The remediation policies were installed to reduce the amount of nitrogen in the river system by local agriculture. The first policy controlled the amount of nitrogen used by the farmer while the second policy taxed the farmer for the amount of nitrogen runoff produced. This research demonstrates the ability of a rural hydrologic decision support tool to aid in rural policies such as agricultural nitrogen usage for the benefit of flora and fauna.

Future Directions

There are many drawbacks for a rural hydrologic decision support system including, cost, technological expertise, data set access, and feasibility. However, data sets are becoming more readily available to the public. Technological advancements have made the usage of these tools easier and more available. More people are cross-educated with GIS, meaning more and more people utilize GIS capabilities in a wide variety of fields. These trends will lead to more rural decision makers having access to the knowledge obtained through the use of GIS and decision support tools.

Cross References

► Decision-Making Effectiveness with GIS
► Distributed Hydrologic Modeling
► Environmental Planning and Simulation Tools
► Hurricane Wind Fields, Multivariate Modeling
► Hydrologic Impacts, Spatial Simulation

Recommended Reading

1. Van Liew, M.W., Garbrecht, J.D., Arnold, J.G.: Simulation of the impacts of flood retarding structures on streamflow for a watershed in southwestern Oklahoma under dry, average, and wet climatic conditions. J. Soil Water Conserv. **58**(6), 340–348 (2003)
2. Schär, C., Vasilina, L., Pertziger, F., Dirren, S.: Seasonal runoff forecasting using precipitation from meteorological data assimilation systems. J. Hydrometeorol. **5**(5), 959–973 (2004)
3. Fohere, N., Haverkamp, S., Frede, H.G.: Assessment of the effects of land use patterns on hydrologic landscape functions: development of sustainable land use concepts for low mountain range areas. Hydrol. Process. **19**(3), 659–672 (2005)
4. Cau, P., Cadeddu, A., Gallo, C., Lecca, G., Marrocu, M.: Estimating Available Water Resources of the Sardinian Island Using the SWAT Model. SWAT2003 2nd International SWAT Conference Proceedings, Bari, Italy. Retrieved July 1, (2004) from SWAT Proceedings Web site: http://www.brc.tamus.edu/swat/2ndswatconf/2ndswatconfproceeding.pdf (64–70) (2003)
5. Melesse, Assefa M., Graham, Wendy D., Jordan, Jonathan D.: Spatially Distributed Watershed Mapping and Modeling: GIS-Based Storm Runoff Response and Hydrograph Analysis: Part 2. J. Spat. Hydro. **3** (2003)
6. Whittaker, G.: Application of SWAT in the evaluation of salmon habitat remediation policy. Hydrol. Process. **19**(3), 839–848 (2005)
7. Echardt, K., Ulbrich, U.: Potential impacts of climate change on groundwater recharge and streamflow in a central European low mountain range. J. Hydrol. **284**(1–4), 244–252 (2003)
8. Grabow, D.: Overland Flooding Risk Assessment in Rural Communities. Master's Thesis, University of North Dakota (2004)
9. Chaplot, V., Saleh, A., Jaynes, D.B., Arnold, J.: Predicting water, sediment, and NO_3-N Loads Under Scenarios of land-use and management practices in a flat watershed. Water, Air, Soil Pollut. **154**(1–4), 271–293 (2004)
10. Gutzler, D., Nims, J.: Interannual Variability of water demand and summer climate in Albuquerque, New Mexico. J. Appl. Meteorology. **44**, 1777–1787 (2005)
11. Sun, H., Cornish, P.S.: Estimating shallow groundwater recharge in the headwaters of the Liverpool Plains using SWAT. Hydrol. Process. **19**(3), 795–807 (2005)

Rural Hydrology

► Rural Hydrologic Decision Support

Sample Trial

► Geosensor Networks, Estimating Continuous Phenonena

Sampling-Based Methods

► Sensitivity Analysis

Scalable Vector Graphics (SVG)

ANDREAS NEUMANN
Institute of Cartography, ETH Zurich, Zurich, Switzerland

Synonyms

SVG; Vector graphics for the Web; XML based vector graphics; Document object model; Web graphics standard; Rich client internet applications; SMIL; Open standard

Definition

SVG is a royalty-free and vendor-neutral two-dimensional web graphics standard developed by the W3C consortium. The XML based markup language describes and integrates vector graphics, raster graphics, text, multimedia, interactivity, scripting and animation. The SVG document consists of a tree-like structure, which can be interpreted by humans and machines. Graphical objects can be grouped, styled, transformed and composited into previously rendered objects. The feature set includes nested transformations, clipping paths, alpha masks, filter effects, template objects and extensibility. As the name indicates, SVG graphics can be scaled while maintaining quality. SVG graphics and applications can be viewed with most modern web browsers natively or by installing a plugin. The SVG standard can be used as a graphics exchange format, as a basis to build interactive and animated applications, web maps and online GIS. Developers familiar with other web standards can benefit from their existing knowledge. Being based on XML, SVG integrates well with modern development tools and common workflows. It also integrates with other W3C and OGC technologies, such as XML, DOM, Scripting, CSS, JPEG, PNG, SMIL, WMS, GML and others. The language independent DOM interfaces allow deleting, creating, changing or reordering elements in the document tree. The most common scripting language used with SVG documents is ECMAScript, also known as Javascript. SVG also supports multi-language documents, font-embedding and internationalization features. The XML namespace support enables the embedding of domain semantics and the extension towards domain specific applications. Network interfaces enable server client communication.

Historical Background

SVG 1.0 was developed by the W3C consortium, the organization responsible for the development of web standards. It became a W3C recommendation in 2001. Many well-known corporations and research institutions contributed to the development of the SVG standard. Version 1.1 was released in 2003 and introduced different modules: SVG 1.1 Full, a module for desktop computers and two SVG mobile profiles: SVG Basic for PDA like devices and Smartphones and SVG Tiny for constrained mobile phones. In parallel, a SVG print module is developed as an XML based alternative to other printing languages, such as Postscript or PDF. It is not yet a W3C recommendation, but SVG is already a widely accepted graphics format in the XSL-FO document generation and printing industry. After the release of SVG 1.1, the W3C SVG working group started work on SVG 1.2. SVG 1.2 Tiny adds, among many other improvements, multimedia support (audio and video), standardized network interfaces, improved text support and a lean Micro-DOM for better scripting support in constrained devices. With mobile devices becoming more and more powerful, SVG Tiny 1.2 is now the only SVG mobile profile.

Although originally designed to be a native web browser technology, the first SVG implementations were either plugins, standalone viewers or programming toolkits. One of the most popular SVG viewing plugins for web browsers was the Adobe SVG viewer (ASV). Apache Batik is an SVG toolkit for Java programmers, a standalone SVG viewer and comes with a number of utilities useful during SVG content development. It is one of the most complete and standard-compliant SVG implementation at the time of writing. The Batik component is also used in the Apache FOP project, an XSL-FO engine. With the rise of native SVG support in popular web browsers (Opera, Mozilla Firefox and Apple's Safari) and Adobe's aquisition of the Macromedia Flash technology (a competing vector graphics and multimedia technology), Adobe decided to stop the development of their SVG plugin. However, at the time of writing, all browser vendors, except Microsoft, worked on implementing or improving the support of the SVG markup language in their browser product, with Opera 9 being the most complete browser implementation. Microsoft confirmed, however, that SVG is an established web standard which they would like to implement. A number of SVG mobile implementations exist: the two most complete commercial implementations are the Ikivo viewer and OpenText Bitflash, followed by the Java based Tinyline implementation. Sun and Nokia developed Java based SVG Tiny toolkits which can be used to display MMS messages, SVG graphics in J2ME applications to integrate SVG support.

After finishing SVG 1.2 Tiny the SVG working group will work on the SVG 1.2 Full profile and continue finalizing the SVG print profile. The SVG specification is developed as part of an open process and built on consensus. The current SVG specifications, working draft and additional information is available at [11]. A working group consisting of W3C members and invited experts regularly meets at telephone conferences and face to face meetings discussing new features, open issues and developer feedback. Any individual can provide feedback and comments on the public W3C SVG mailing list. It is obvious that such an open process takes longer than the development of proprietary de facto graphics standards that only need to work in the products of a single company, but it is hoped that the public feedback and review process creates a better graphics standard for the long run.

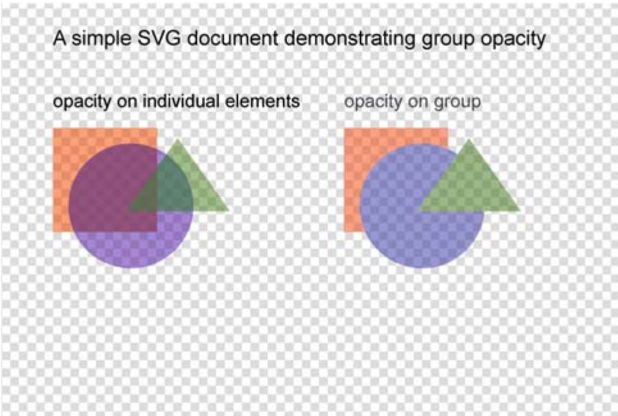
Scientific Fundamentals

Painters Model and Document Structure

The rendering of the treelike document structures of SVG documents follows the simple and straight-forward painters model, with an implicit drawing order. Elements appearing first in the document tree (higher up in the file) are rendered first, subsequent elements that potentially overlap with previous elements are drawn on top of them, taking compositing rules into account. Elements in the document tree can be grouped (<g/> element). Groups may be transformed, hidden or made visible, and share common presentation attributes. Individual elements can still overwrite centrally defined attributes. To improve human readability and semantic processing of documents, metadata elements, title and description elements may be included at any place in the document tree. SVG documents may also contain a <defs/> section for the central definition of re-usable elements and definitions, such as geometry, pattern definitions, gradient definitions, symbol definitions, etc. Elements in the <defs/> section aren't rendered, only their instances. Two attributes, *display* and *visibility*, define whether an element or group of elements is visible or not. The display attribute removes the elements temporarily from the rendering tree and inherits into child elements, the visibility attribute just hides the elements, but they may still be sensitive to events. Child elements may override the visibility attribute of its parent. Every element can have unique ids which can be used for later reference or script access. Figure 1 demonstrates the structure of a simple SVG file with some basic shapes, the use of the <title/> element, grouping, the difference between simple opacity and group opacity and the use of patterns.

Coordinate Systems, Units and Transformation, Clipping, Masking and Compositing

The initial viewing area of an SVG document is called the *viewport*. Its dimensions are defined through several methods: width and height of the browser window (in case of a directly viewing of an SVG file or when using SVG as a CSS background), width and height of the <embed/>, <iframe/> or <object/> element where the SVG file is embedded (e. g. within a HTML file) or CSS positioning properties. Additionally, the <svg/> root element may have attributes like *width*, *height* (with optional units) and *viewBox*. If the SVG graphics should adopt to various viewport sizes it is recommended to set the *width* and *height* attribute to 100% and specify a *viewBox* that defines the internal coordinate system of the documents canvas. The *viewBox* attribute can also be used to clip content larger than the viewBox or for zooming and panning of the graphics. <svg/> elements can be nested and each nested <svg/> element can define its own coordinate system. For example it is quite common to establish a screen-oriented coordinate system for the UI of an interactive mapping application and to define one or more nested <svg/> elements containing map coordinate systems (e. g. based on




```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
 "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
<svg xmlns="http://www.w3.org/2000/svg" width="100%" height="100%"
 viewBox="0 0 600 400">
  <title>A simple SVG document demonstrating group opacity</title>
  <defs>
   <pattern id="checkerBoard" patternUnits="userSpaceOnUse" x="0" y="0" width="20"
      height="20" viewBox="0 0 20 20">
           <rect x="0" y="0" width="10" height="10" fill="rgb(220,220,220)" />
           <rect x="10" y="0" width="10" height="10" fill="white" />
           <rect x="0" y="10" width="10" height="10" fill="white" />
           <rect x="10" y="10" width="10" height="10" fill="rgb(220,220,220)" />
   </pattern>
  </defs>
  <rect fill="url(#checkerBoard)" x="0" y="0" width="600" height="400" />
  <text x="50" y="40" font-size="20">A simple SVG document demonstrating
   group opacity</text>
  <g id="normalGroup" transform="translate(0,20)">
           <title>Normal Group with opacity on single elements</title>
           <text x="50" y="80" font-size="18">opacity on individual elements</text>
           <rect x="50" y="100" width="100" height="100" fill="red" opacity="0.5" />
           <circle cx="125" cy="175" r="60" fill="blue" opacity="0.5" />
           <polygon points="120,180 170,110 220,180" fill="green" opacity="0.5"/>
  </g>
  <g id="opacityGroup" transform="translate(280,20)" opacity="0.5">
           <title>Group with group opacity</title>
           <text x="50" y="80" font-size="18">opacity on group</text>
           <rect x="50" y="100" width="100" height="100" fill="red" />
           <circle cx="125" cy="175" r="60" fill="blue" />
           <polygon points="120,180 170,110 220,180" fill="green" />
  </g>
</svg>
```

Scalable Vector Graphics (SVG), Figure 1 A simple SVG file demonstrating grouping, group opacity, basic shapes and patterns. The graphics is on the *left side*, the corresponding source code of the graphics on the *right side*

meters). Metadata information on geographic coordinate systems and projections may be embedded in SVG using RDF and OGC syntax.

Regarding transformations, SVG supports translation, scaling (also independent for the x and y axis), rotation (also around a specified center), skewing and matrix transformation. Matrices are very convenient when more complex transformations and calculations are necessary. Transformations and <svg/> elements may be nested as many times as required. The outline of any basic shape, <path/> elements and text may serve as a clipping path or mask or may be clipped by another clipping path. While clipping paths are *hard masks*, one can define individual opacity settings for every part of a *mask*, also using gradients in the alpha channel.

Basic Geometry Elements

SVG supports rectangles (<rect/>), circles (<circle/>), ellipses (<ellipse/>), lines (<line/>), polylines (<polyline/>) and polygons (<polygon/>) as basic shapes, and the more powerful and complex <path/> element for modeling any arbitrary 2D shape. All basic shapes may be replaced by <path/> elements, but the basic shapes exist for convenience in hand-coding SVG files, for semantic reasons and because basic shapes are often easier to animate. The <path/> element supports straight lines, arc segments, quadratic and cubic splines, multiple polygons within a single element and holes in polygons. To compress the geometry definition one can use relative coordinates (relative to the previous coordinates) or shortcuts, such as the horizontal or vertical *lineTo* commands. The path syntax is very similar to the path syntax in Postscript and PDF. The syntax of both basic shapes and the <path/> element is straightforward for both humans and machines. The source code of some basic shapes is demonstrated in Fig. 1.

Text and Fonts

SVG supports almost any text feature available in modern graphics and DTP software. This includes all sorts of text decoration and formatting options, text aligned on path, rotated glyphs, control of spacing, etc. The text on path feature enables the creation of complex line styles by placing individually designed glyphs along lines. SVG 1.1, unfortunately, does not support flow text. This missing feature is now defined in SVG 1.2 tiny and in a more powerful version in SVG 1.2 Full. In SVG 1.1 simple text flow can be realized using Javascript. SVG is also well prepared for multi-language applications and internationalization. It supports Unicode, right to left, top to bottom and bidirectional text. Multi-language documents may be realized using the switch statement and the browser language settings. SVG fonts allow the embedding of fonts within SVG documents, or centralized in external resource documents. This ensures precise and consistent layout across multiple browsers and operating systems. The geometry of


```
<?xml version="1.0" encoding="UTF-8"?>
<svg width="300" height="600" viewBox="0 0 300 600" xmlns="http://www.w3.org/2000/svg"
  xmlns:xlink="http://www.w3.org/1999/xlink" xml:space="preserve" version="1.1"  baseProfile="full">
<title>Demonstration of a few text features</title>
<defs>
 <style type="text/css">
  <![CDATA[
   .red {fill:red;}
   .blu {fill:blue;}
   .darkorg {fill:darkorange;}
   .gra {fill:grey;}
   .sser {font-weight:normal; font-family: sans-serif; text-decoration:none;}
   .seri {font-weight:normal; font-family: serif; text-decoration:none;}
  ]]>
 </style>
 <clipPath id="pix">
  <text x="15" y="460" font-weight="bold" font-size="55" font-family="sans-serif">clipping</text>
 </clipPath>
</defs>
<g id="mainlayer" font-size="20">
 <text class="sser" x="44" y="30">City</text>
 <text class="seri" x="140" y="30">County</text>
 <text class="sser gra" x="32" y="66" transform="rotate(20)"><tspan dx="3 3 3 3 3 3 3 3 3">Street</tspan></text>
 <text class="sser red" x="140" y="73">lines:<tspan x="140" y="98">first line</tspan>
    <tspan x="140" y="123">2nd line</tspan></text>
 <text x="10" y="165" class="sser">The color is <tspan class="darkorg">now darkorange.</tspan></text>
 <text class="sser" x="10" y="205" fill="rgb(255,164,0)" letter-spacing="10">letter-spacing</text>
 <text x="10" y="255" class="seri">A <tspan dy="-9"> higher</tspan>
    <tspan rotate="10 35 50 65 80" dx="0 5 9 9 12">Word!</tspan></text>
 <path id="MyPath" d="M 30 330 s 50 -50 100 0 s 70 -30 110 0" stroke="blue" fill="none" stroke-width="10"/>
 <text class="sser blu">
            <textPath xlink:href="#MyPath" startOffset="5%"><tspan dy="-10">The Times</tspan></textPath>
 </text>
 <text class="sser darkorg" x="15" y="389" font-size="40" stroke="blue" stroke-width="1">Borders ...</text>
 <g clip-path="url(#pix)">
  <image x="10" y="410" width="290" height="290" xlink:href="fill.png"/>
 </g>
</g>
</svg>
```

Scalable Vector Graphics (SVG), Figure 2 Demonstration of some Text Features and Styling with CSS. The graphics is on the *left side*, the corresponding source code of the graphics is on the *right side*

individual <glyph/> elements may contain either path syntax (*d* attribute of the <path/> element) or arbitrary geometry, and even animations. Figure 2 demonstrates some text features: the use of the <tspan/> element for linebreaks and different formatting of portions of a text, the letter-spacing attribute, moving and rotating individual glyphs, text on path, text with stroke and clipping.

Filling and Stroking Options, Markers

SVG supports filling and stroking with solid colors, gradients (linear and radial) and patterns (vector based, raster based). Figure 1 demonstrates the use and definition of a vector based pattern (checkerboard). Stroking options include the specification of stroke-width, stroke-linecap, stroke-linejoin, miter-limit and dashing. Opacity may be specified separately for fill and stroke. The <marker/> elements are in particular interesting for cartography and GIS: a marker is arbitrary geometry placed at each vertex of a <path/> element or basic shape. This feature contributes to the definition of more complex line styles. An example for the use of markers are specific line types, where the marker symbols represent poles. Markers are also used for attaching arrows to lines. Markers may be automatically oriented according to the line or segment angle they are attached to. All stroke and fill attributes, including gradient and pattern definitions may be animated. Fill rules define how to deal with holes in paths: *nonzero* (default) only creates holes if the path direction of the inner polygon winds in the opposite direction than its parent path, while *evenodd* always creates holes for inside paths.

Filter Effects

Filters are a very powerful feature in SVG, enabling many *special effects*. Filters can be attached to both raster and vector elements. Vector elements are rasterized during the rendering pipeline, hence there is an opportunity to include filters. Typical applications for filters are color corrections, brightness and contrast adaptations, blurring and sharpening, illumination filters, generation of drop shadows and halo effects, convolution filters, displacement and morphology filters, generating turbulence, etc. Filters may be *pipelined* (combined) in any order. The output of one filter may be piped to the input of the next filter. Every filter parameter can be animated which can lead to very interesting effects. Filters are very powerful visualization options, but, depending on the complexity or size of the objects, may require a fair amount of computing power. Figure 3 demonstrates the use of SVG filters for shadow and lighting effects, a combination of a gaussian blur, a lighting and compositing filter.


```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
        "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="7.5cm" height="5cm" viewBox="0 0 200 120"
   xmlns="http://www.w3.org/2000/svg" version="1.1">
 <title>Example filters01.svg - introducing filter effects</title>
 <desc>An example which combines multiple filter primitives to produce a 3D lighting
          effect on a graphic consisting of the string "SVG" sitting on top of oval filled in
          red and surrounded by an oval outlined in red.</desc>
 <defs>
  <filter id="MyFilter" filterUnits="userSpaceOnUse" x="0" y="0" width="200" height="120">
   <feGaussianBlur in="SourceAlpha" stdDeviation="4" result="blur"/>
   <feOffset in="blur" dx="4" dy="4" result="offsetBlur"/>
   <feSpecularLighting in="blur" surfaceScale="5" specularConstant=".75"
     specularExponent="20" lighting-color="#bbbbbb" result="specOut">
    <fePointLight x="-5000" y="-10000" z="20000"/>
   </feSpecularLighting>
   <feComposite in="specOut" in2="SourceAlpha" operator="in" result="specOut"/>
   <feComposite in="SourceGraphic" in2="specOut" operator="arithmetic"
         k1="0" k2="1" k3="1" k4="0" result="litPaint"/>
   <feMerge>
    <feMergeNode in="offsetBlur"/>
    <feMergeNode in="litPaint"/>
   </feMerge>
  </filter>
 </defs>
 <rect x="1" y="1" width="198" height="118" fill="#888888" stroke="blue" />
 <g filter="url(#MyFilter)" >
   <path fill="none" stroke="#D90000" stroke-width="10"
      d="M50,90 C0,90 0,30 50,30 L150,30 C200,30 200,90 150,90 z" />
   <path fill="#D90000"
      d="M60,80 C30,80 30,40 60,40 L140,40 C170,40 170,80 140,80 z" />
   <text fill="white" stroke="black" font-size="45" font-family="Verdana" x="52" y="76">
          SVG</text>
 </g>
</svg>
```

Scalable Vector Graphics (SVG), Figure 3 SVG Shadow and Lighting Filters. Source: [3]

Styling

There are several methods to style SVG elements. One is to use *presentation attributes* directly in the elements. These attributes are inherited if specified on a <g/> element. This method is preferred if the attributes have to be frequently modified using scripting and the DOM. It is also the only method available in SVG mobile. The second method is to use external or internal CSS definitions and a *class* attribute on the elements. The advantage is a centralized handling of styles and a quick adaptation to a different look and feel. The third method is to use the *style* attribute and define all styling parameters in a semicolon separated list of styling parameters. Aural and media styles are also supported, but at the time of writing not in all user agents. Figure 2 demonstrates the use of CSS and presentation attributes for styling.

Interactivity and Scripting

Interactivity and scripting are key parts when it comes to making SVG appealing for developing rich client internet applications (RIA). SVG graphics are by default zoomable and pannable. Many SVG viewers support additional interactions, such as search for text, or start/pause animations. SVG supports hyperlinks, display of tooltips and custom cursors. Interactivity is enabled by events, a scripting language and SMIL. SMIL is a declarative way of specifying interactions or animations. SMIL constructs generally contain a trigger (either time based or event based), the target, the attribute to change, duration and interpolation parameters. SMIL can be regarded as a simple, declarative scripting language.

Various event types enable script or SMIL operations in reaction to user or system events. Table 1 summarizes the available events in SVG 1.1. Any of the events can trigger either a script function or a SMIL interaction. Mutation events listen to changes within a particular node in the XML document tree.

A more powerful and flexible way to modify SVG documents is the use of a client-side scripting language. Scripts can either be embedded in the SVG files or referenced (external files). SVG defines a language independent API to access and manipulate the SVG DOM. The most widely used and implemented scripting language in conjunction with SVG is ECMAScript (the standardized version of Javascript). One reads or changes attributes, creates, moves or deletes elements and loops over the document tree as it is the case with any XML or XHTML document. SVG also provides network interfaces to directly talk to server-side applications. .getURL() and .postURL() are methods which allow transfering and retrieving of data from and to the server without having to reload the SVG file. SVG 1.2 adds additional network options for client-server communication.

A very useful attribute regarding interactivity is the *pointer-events* attribute. This attribute controls the sensitivity of graphic elements regarding the reaction to mouse events. One can either set the attribute to *none* (no reaction to

Scalable Vector Graphics (SVG), Table 1 SVG Events

Status Events	Zoom and Scroll Events	UI Events	Mouse Events	Keyboard Events	Animation Events	Mutation Events
SVGLoad	SVGResize	focusin	click	keydown	beginEvent	DomSubtreeModified
SVGUnload	SVGScroll	focusout	mousedown	keyup	endEvent	DomNodeInserted
SVGAbort	SVGZoom	activate	mouseup	keyup	repeatEvent	DomNodeRemoved
SVGError			mouseover			DomNodeRemovedFromDocument
			mousemove			DomNodeInsertedIntoDocument
			mouseout			DomAttrModified
						DomCharacterDataModified

mouse events), *fill* or *stroke* and some additional options. By using this attribute one can avoid that an element above another element "steals" the event. In web mapping and online GIS applications, the pointer-events attribute helps to control which layer receives mouse events for the display of attribute values associated with map graphics.

Animation

Almost any element and attribute can be animated in SVG. There are currently two ways to implement animations in SVG: the first method is to use Javascript and a timer and to repeatedly change attribute values in elements. This approach requires programming know-how but guarantees maximum flexibility when it comes to interpolation methods and logic. The second and often easier way is again SMIL, a descriptive way to define animation parameters. SMIL animations can trigger script execution and vice versa. Both, script based and SMIL animations can be triggered by the events listed above.

SMIL offers five elements for descriptive animations: <animate />, the most general element for animating numeric, interpolateable attributes, <set /> for setting non interpolateable attributes, such as string-based values, <animateMotion /> for moving elements along a motion path, <animateColor /> for animating color values and finally <animateTransform /> for animating transform attributes. Common attributes of the five animation elements are *begin*, *end*, *dur* (duration), *from*, *to*, *by*, *repeatCount*, *repeatDur*, *fill*, *calcMode*, *keyTimes*, *values* and *keySplines*. Of particular interest are the latter attributes: *calcMode* allows to specify the interpolation method (discrete, linear, paced and spline), *keyTimes* and *values* allow the setting of timestamps (in percentage of the full duration) and corresponding values, fixpoints that the interpolation has to respect, and *keySplines* define acceleration or deceleration effects. In interactive maps *keySplines* can be used to present acceleration effects and speed of moving objects, such as vehicles or airplanes.

Extensibility

As a XML based language, SVG supports foreign namespaces. It is possible to define new elements or add new attributes to existing SVG elements. Elements and attributes in a foreign namespace have a prefix and a colon before the element or attribute name. Elements and attributes in foreign namespaces that the SVG viewer does not know, are ignored. However, they can be read and written by script. Foreign namespaces are used to introduce new elements (e. g. GUI elements, scalebars) and for the attachment of non-graphical attributes to SVG graphic elements (e. g. GIS non-graphical attributes). Those attributes can be analyzed and used to create thematic maps or charts. Additionally, one can display those attributes upon mouseover. This is not only useful for maps and drawings, but also for user-interfaces, technical drawings, charts, etc. For the inclusion of metadata, the W3C consortium recommends the use of the RDF (resource description framework) or the Dublin core standard. The RDF fragments should be included in a <metadata /> tag and should be defined in a foreign namespace.

Key Applications

Mobile SVG and Location Based Services

SVG is already implemented in the majority of mobile phones available today. SVG is part of the 3GPP mobile phone standard. Older implementations only support animation and SMIL based interactivity, newer versions also support scripting, video and audio. Applications of SVG mobile are *location based services*, *animated or interactive MMS messages*, *business graphics and charts*, *games* and many more.

Graphics in Web Browsers

The majority of graphics in today's webpages is still in raster format (jpeg, png, gif) or in proprietary vector formats requiring additional browser plugins. While raster graphics work fine for photos, it's not the best choice for drawings, CAD files, presentation graphics, animated graphics, interactive graphics, etc. Adding interactivity to traditional DHTML and raster graphics requires a lot of ugly workarounds. Most of these applications are more elegantly handled by SVG graphics.

GIS and Web Cartography

The rich graphic features and effects, as well as the support of nested cartesian coordinate systems, extensibility through XML namespaces and the rich interactivity and multimedia features, make SVG a good choice for interactive mapping systems. With the solid support of vector graphics, raster graphics and text features, it already satisfies many of the requirements of modern web mapping. Missing features, such as complex line styles can partially be compensated by the use of server side generation or client side scripting. The XML base of SVG enables straight-forward conversion from other XML based OGC standards, such as GML (Geography Markup Language). A number of GIS, map server or graphics software already supports the export of SVG, however, not always in optimized form. Several commercial web mapping services, such as Google Maps and Microsoft Virtual Earth, use SVG for the drawing of vector based overlay elements on top of their raster based map tiles, e. g. to display routing information.

The scripting and event support, combined with network interfaces allows for efficient client-server applications which can even be used for distributed data acquisition purposes (e. g. online digitizing of GIS features). The [6] offers many tutorials, examples and ideas for interactive web cartography, web mapping and online GIS applications based on SVG. It is worth mentioning that large vector based geographic databases can't be visualized without data filtering and generalization. When publishing large GIS data sets on the web, the use of spatial databases proved to be valuable. These databases allow geographic queries, filtering based on attributes and partially on-the-fly generalization or simplification. The article *Dynamic Loading of Vector Geodata* [7] discusses the generation of SVG maps from spatial databases and various optimization strategies.

Print and Reporting

A separate SVG print module exists, which is a royalty free XML based page description language alternative to Postscript or PDF. It extends the SVG base language with additional printing features, such as a <page/> element, paper size control, device color specifications, overprinting and color management support. At the time of writing, this profile is still under development and not yet available in printing devices. The current working draft of the SVG print specification can be found at [12]. It is also worth mentioning that SVG graphics can be used within XSL-FO (Formatting Objects) documents to incorporate vector graphics into page content. SVG is supported by most XSL-FO processors.

E-Learning and Multimedia

SVG's rich visualization options and the support of interactivity make it a natural candidate for providing graphics and interactive examples in e-learning environments. The ability to access the SVG source code and have a glance "under the hood" to see how things are made, is especially useful for learning and sharing purposes. SVG also provides a fun way to introduce programming and illustrate the functionality of algorithms. Students are usually motivated if they can graphically visualize what they program. Examples for interactive and animated E-Learning content based on SVG can be found at [1,4,8].

Design and CAD, Export and Exchange Format

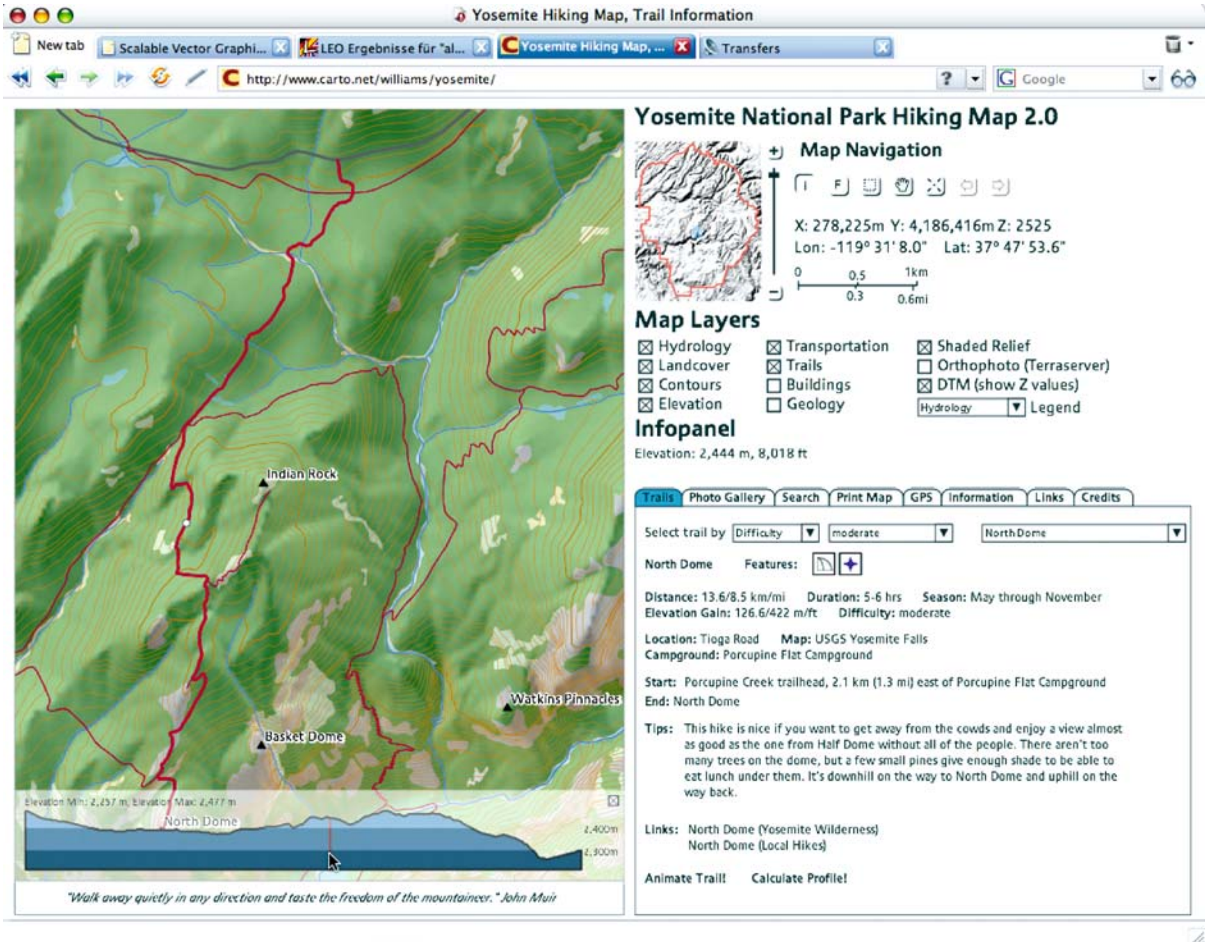
One of the original goals when designing the SVG specification was the establishment of SVG as an exchange and export format for graphics design, CAD and charting software. The current state is that a large number of applications can generate SVG content but few can properly import SVG documents with all features present in the current SVG 1.1 full specification. Some graphics software, such as Inkscape [5] use SVG as a native file format.

Web Applications, Rich User Interfaces

SVG can be used as a platform upon which to build graphically rich internet applications (RIA) and user interfaces. With SVG, the application developer can use a stack of open standards. They are not tied to one particular implementation, vendor or authoring tool. On the server, developers can tie into any server side programming language or framework they are familiar with. Many client side frameworks, such as the Dojo [2] or other Javascript/Ajax Toolkits partially build on SVG (or VML on Internet Explorer) or translate SVG into other frameworks. A number of SVG based GUI elements are available at [9].

Embedded Systems

SVG is well suited as a graphical frontend (GUI and information graphics) in embedded devices, e. g. GPS devices, car navigation systems, for displaying the status of a machine or industrial device or for monitoring sys-



Scalable Vector Graphics (SVG), Figure 4 Yosemite Hiking Map – an example SVG web mapping application. Source: [13]

tem status and controlling device functionality. Many SVG mobile viewers already run in resource constrained devices and on multiple operating systems suitable for embedded devices.

Future Directions

While SVG is still under development (both in specification and in SVG UA's) it can already be used in serious projects, such as mapping portals and online GIS, as web mapping projects like Google Maps, Microsoft Virtual Earth and the examples on carto.net successfully demonstrate. SVG support is rapidly developing both in desktop and mobile browsers. Widespread use of SVG in desktop web browsers will likely start once it is implemented in Microsoft Internet Explorer, without the requirement of an additional plugin. Upcoming features in the SVG 1.2 Full specification will introduce a lot of appealing additional features useful for mapping: vector effects, better compositing, better text flow, editable text, multi resolution images, progressive rendering and streaming support, transition effects and more. The SVG Print activities will enable and simplify printing workflows. It is obviously easier and less vendor dependent to generate, convert and manipulate XML based documents than Postscript or PDF files.

Cross References

- ► Geography Markup Language (GML)
- ► Internet GIS
- ► Mobile Usage and Adaptive Visualization
- ► Multimedia Atlas Information Systems
- ► OGC's Open Standards for Geospatial Interoperability
- ► PostGIS
- ► User Interfaces and Adaptive Maps
- ► Vector Data
- ► Web Mapping and Web Cartography

Recommended Reading

1. Dagan, S.: Math Animated – A hosted teaching aid using graphic animations and interactivity (2005). http://www.mathanimated.com/. Accessed 12 Nov 2006
2. DOJO: DOJO – the javascript toolkit (2005). http://dojotoolkit.org/. Accessed 12 Nov 2006
3. Ferraiolo, J., Fujisawa, J., Jackson, D.: Scalable Vector Graphics (SVG) 1.1 Specification (2003). http://www.w3.org/TR/SVG11/index.html. Accessed 19 Nov 2006
4. Hirtzler, M.: Pilat Informative Educative: SVG – PHP – MySQL – JavaScript (2000–2006). http://pilat.free.fr/english/index.htm. Accessed 12 Nov 2006
5. Inkscape Community: Inkscape, SVG Authoring Tool (2000–2006). http://www.inkscape.org. Accessed 12 Nov 2006
6. Neumann, A., Winter, A.M.: Carto.net SVG page: articles, examples and tutorials (2000–2006). http://www.carto.net/papers/svg/. Accessed 12 Nov 2006
7. Neumann, A.: Dynamic Loading of Vector Geodata for SVG Mapping Applications Using Postgis, PHP and getURL()/XMLHttpRequest() (2005). http://www.carto.net/papers/svg/postgis_geturl_xmlhttprequest/. Accessed 12 Nov 2006
8. Neumann, A.: Use of SVG and ECMAScript Technology for E-Learning Purposes, in: Proceedings of the ISPRS Workshop Commissions VI/1 VI/2, Tools and Techniques for E-Learning, Potsdam, Germany, June 1–3 (2005)
9. Neumann, A.: SVG GUI (2003–2006). http://www.carto.net/papers/svg/gui/. Accessed 12 Nov 2006
10. SVG Community: SVG.Org – an SVG community page, news and links for SVG (2004–2006). http://www.svg.org/. Accessed 12 Nov 2006
11. W3C: Scalable Vector Graphics (SVG) – XML Graphics for the Web (1999–2006). http://www.w3.org/Graphics/SVG/. Accessed 12 Nov 2006
12. W3C: SVG Print Specification (2003–2006). http://www.w3.org/TR/SVGPrint/. Accessed 12 Nov 2006
13. Williams, J.: Yosemite Hiking Map 2.0 (2006). http://www.carto.net/williams/yosemite/. Accessed 12 Nov 2006

Scale Dependencies

► Scale, Effects

Scale, Effects

TEMUULEN TSAGAAN SANKEY
GIS Training and Research Center,
Idaho State University, Pocatello, ID, USA

Synonyms

Scale rendering; Scale dependencies

Definition

The term "scale" refers to the dimensions of an observation. Spatial scale describes both the geographic extent and the resolution of a study. It can also refer to a cartographic scale, which is a ratio of length on a map to a real-world distance. The main issue related to scale is that data, models, or maps at different scales convey different sets of information. Data on a map with a large scale show a lot of detailed information, whereas a map with a small scale can only show general information and has no detail. A town map can show streets and business locations in town, whereas a map of the USA does not show this kind of detail. Therefore, details of information are lost when changing from a large scale to a small scale. This is known as "the effects of scale". Another example is that a person can study a forest, or a tree, or a leaf. The information obtained can be very different depending on the scale of the study. Consequently, models of a leaf would not be directly applicable to a tree or a forest and the scale difference needs to be carefully dealt with before transferring information from one scale to another. Other related issues to scale are determining what variables or features change across scales, measuring the scale effects, identifying an appropriate scale for a study, integration of scale into analysis, aggregation and disaggregation across scales and their effects on results, and implementation of a multiscale study. Issues of scale are particularly important in ecology, geography, and the social sciences. In geographic information systems (GIS) environments, different tools have been developed to retain information when zooming out from one scale to another. Scientists still have to decide what level of detail should be included at what scale and how to incorporate effects of scale into analyses.

Historical Background

Scale has always been an important topic in geography and other related disciplines. Cartography and mapmaking in France and other European countries initially started the development of scale thinking and cartographic scale during the eighteenth century [1]. Since then many different disciplines and subdisciplines have developed their own set of theories and methods regarding scale and its effects. In particular, over the last few decades, theories of modifiable areal unit problems (MAUPs) have been developed in geography, the hierarchy theory in ecology, and the ecological fallacy in human and physical geography [1]. The increasing use of remote sensing and geographic information systems (GIS) in recent decades has attracted even greater attention from the scientific community to the effects of scale. The abundant diversity of satellite remote sensing imagery and GIS data increases the availability of multilevel, multiscale, and multitemporal data. For example, scientists now often face decisions of scale and choices of satellite imagery. High-resolution satellite imagery from the Quickbird platform (2.4 m cell size) shows a lot more detail than National Oceanic and Atmospheric Administration (NOAA) Advanced Very High Resolution Radiometer (AVHRR) imagery with 1 km cell size, but a Quickbird image scene covers a much smaller area than a NOAA AVHRR scene. The current demand in analysis techniques for multiscale data demonstrates the need for further development of theories and methods for dealing with the effects of scale.

Scientific Fundamentals

The main issue related to scale is that different levels of detail are presented at different scales. As the scale becomes smaller, the level of detail decreases. In GIS environments, different tools and applications have been developed to deal with this issue. In GIS, it's possible to zoom in and out and thus change the scale of the map display. When there are multiple layers in GIS, all layers can be displayed simultaneously and zooming in and out of them is possible. However, on zooming out from large to small scales, some of the layers with a lot of detailed information become clustered and no longer display the details of the information. To deal with this issue, a map maker can decide what level of detail should be displayed at what scale. This is known as rendering. This tool allows a GIS user to set a scale range and display certain layers only when the map is within the specified range of scales. National map framework standards and the Federal Geographic Data Committee also provide guidelines regarding the content and level of detail for different maps at various scales. A scale dependency can be set for maps according to these standards. The scale-dependency technique allows users to change scales from one hierarchical level to another with a single click of a mouse. For example, using a 12-click scale-dependency technique, a map can be zoomed in 12 hierarchical levels or 12 different scales. A map scale can be changed with a single click from a state map to a map of multiple counties and the next click would zoom into a single county and so on.

In addition to the GIS tools and the applications described above, theoretical aspects of scale effects have been developed to address the issues related to scale. The theories also deal with the fundamental issue of varying information at varying scales. Scientists study the earth system and its residing populations to better understand its physical and biological characteristics. Most physical and biological variables, however, function across a vast array of scales from microbial to global and are challenging to examine at all scales simultaneously. It's necessary, therefore, to study earth phenomena at particular scales and extrapolate the information to other scales. Unfortunately, the ability to do so is limited by the nature of the phenomena, namely, scale dependency. Most earth processes and patterns are scale dependent [2]. In other words, their characteristics vary when considered at different scales and thus prevent extrapolation across scales. This is also commonly referred to as the "effects of scale" and has important implications for spatial data analysis, modeling, interpretation, and inferences [2]. Certain aspects of the effects of scale have been described as the MAUP and the ecological fallacy problem. Other aspects of the effects of scale include problems in identifying the appropriate scale and optimum methods of analysis, and limitations on inferences.

The MAUP was first termed by Stan Openshaw in 1984 [3], although it had been long recognized by scientists. MAUP describes two distinct properties of spatial data. The first property is that the results of spatial data analysis can vary, when spatial data are analyzed in different aggregates or areal units. For instance, the estimated correlation coefficient between two variables can vary, when the two variables are analyzed in different aggregates. One study estimated the correlation coefficient between the percentage of native born population and percentage of illiteracy to be 0.118, when analyzed at the individual level, but –0.619 when analyzed at the census division level [3]. Such effects have important implications for the inferences and interpretation of spatial processes and patterns. The inferences and interpretation are seriously limited by the feasibility and validity of extrapolation. One needs to be careful with the kinds of conclusions to make when presented with a single result from a single spatial scale because the result might indicate an opposite, or at least different, pattern at another scale. The second property is that the results of spatial data analysis can vary, even at similar scales, depending on how the data is aggregated. In other words, the method of spatial data aggregation can have an important effect on the results of spatial data analysis. Possible aggregation methods used in GIS include the sum, mean, median, maximum, and minimum values of the smaller areal units to be aggregated. The use of mean versus maximum values of the finer areal units, for example, might reveal different spatial patterns. This also has important implications for the inferences and interpretation of spatial processes and patterns. The properties described by MAUP suggest that spatial data analyses involving aggregates, areal units, and pixels have important considerations and decisions to make regarding the scale, resolution, and aggregation methods to use.

The ecological fallacy problem was first identified in 1969 by Hayward R. Alker, who defined three specific issues related to inferences made from spatial analyses [3]. The first issue is termed the individualistic fallacy, in which inferences about fine-scale properties are made erroneously from coarse-scale properties. This problem is common in many disciplines where observations are made at the individual level at fine scales and inferred to population levels at coarser scales. This problem can also occur when observations are made at the lower level of a process operation such as the microbial level, and inferences are made regarding the higher level such as the global level. The second issue is termed cross-level fallacy, in which inferences are made from one population about another at the same scale. In this case, inferences can be erroneous not due to scale differences, but due to the nature of the process or pattern under investigation in terms of its repeatability and feasibility for extrapolation. For example, children's reading skills at age five in one country might not be the same in a different country. Therefore, one can not extrapolate the results of one study in one country to other countries. The third issue, the ecological fallacy, arises when erroneous inferences are made about finer scale processes from coarser scale observations. This is the opposite of the individualistic fallacy and can occur in studies where observations are made at the population level, but inferences are made regarding the individual level.

Inferences of spatial processes and patterns are limited by the nature of the phenomenon under investigation, specifically in terms of its repeatability. Some spatial processes might operate at certain scales only, while other processes might operate at multiple scales and across scales. Similarly, some spatial patterns are observed at certain scales only, while other patterns might hold the same at multiple scales. For instance, tree distribution in a savanna environment might appear clustered at one scale, while it might appear random or uniform at another scale. The observed pattern depends upon how close the analysis is zoomed in or out. This means that some spatial processes and patterns need to be observed at multiple scales and each scale reveals a different process and a different pattern. Consequently, inferences across scales on these processes and patterns can not be made easily.

These problems demonstrate the need for identifying an appropriate scale of analysis. In other words, an appropriate area with an associated size, shape, and resolution needs to be determined. This largely depends upon the hierarchical level at which a spatial process operates. Photosynthesis and transpiration, for example, occur at a leaf level. These processes, therefore, could be observed at a measurement scale of millimeters to centimeters (or a meter at most). Identifying the appropriate measurement unit requires identification of the underlying process and the levels at which this process might operate. Some scientists suggest that the hierarchy theory provides a useful framework for this purpose, although some geographers do not readily accept this idea. In hierarchy theory, a system can naturally be subdivided into two dimensions: a vertical structure of levels and a horizontal structure of "holons" or "wholes" at the same level [1]. Interactions occur both among the wholes within each level and across the hierarchical levels within the vertical structure. This suggests that spatial processes and the associated patterns can be divided into distinct temporal and spatial scales that correspond to the hierarchical levels. This, in turn, implies that spatial processes and patterns have characteristic spatial and temporal scales associated with them. These scales are referred to as "operational scales". In terms of the interactions between levels, the hierarchy theory suggests that processes at a given level are constrained by the next higher level, but their mechanisms are driven by the next lower level. In other words, lower levels reside within larger levels. This relationship provides a useful framework for multiscale analysis and monitoring. In terms of quantitative approaches, a method of local variance has been proposed for identifying an appropriate scale at which to study landscape patterns using remote sensing imagery.

The scale-dependent nature of a spatial process or a pattern can often be site-specific or time-dependent. It is therefore challenging to identify general methods to use in all studies intended for detecting effects of scale. Several methods, however, are commonly proposed [3]. Among them, the simplest method might be a plot of the change in variance and the different sizes of possible measurement units. If there is no scale dependence, the resulting plot should have a straight line with a slope of –1.0 when expressed as:

$$\ln S^2 = a - \ln n$$

where S^2 is the variance and n is the size of the measurement unit. This method is very similar to the geographic variance method, in which one can use an analysis of variance (ANOVA) approach to determine the level at which the highest variance is observed [3]. Once the total sum of squares for the data is estimated, it can be divided into portions associated with each spatial scale. The scale at which the largest portion of the total variance is observed represents the scale at which a spatial process of interest operates.

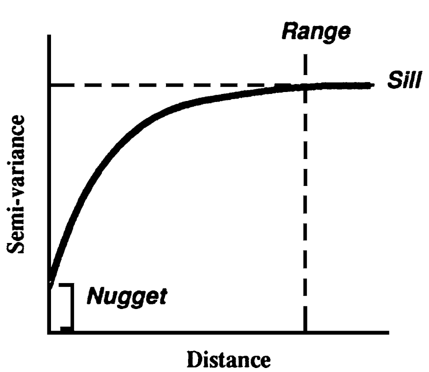
A similar approach, the local variance method, is also based on the calculation of the largest variance. This method was developed for remote sensing imagery analysis and uses a 3×3 moving window [3]. The moving window estimates the average value of the variances within the window, known as the local variance. The local variance is high when the window size matches the optimum scale at which the spatial pattern of interest should be observed. However, the local variance is low, if the spatial pattern of interest should be observed at a coarser scale than the window size. The local variances can also be plotted against progressively increasing sizes of the moving window. The highest point in the plot indicates the scale at which the highest local variance is observed.

A closely related method to the local variance method is the texture analysis method [3]. Texture can be analyzed within a 3 × 3 moving window using different indices of minimum and maximum values, standard deviation, and local variance. These indices describe the variation in the digital values of a raster image. Higher textural index values indicate greater heterogeneity, while lower index values represent homogeneity at the scale of the moving window. The indices can be estimated a number of times with progressively increasing sizes of the moving window to describe the spatial patterns throughout the image at different scales.

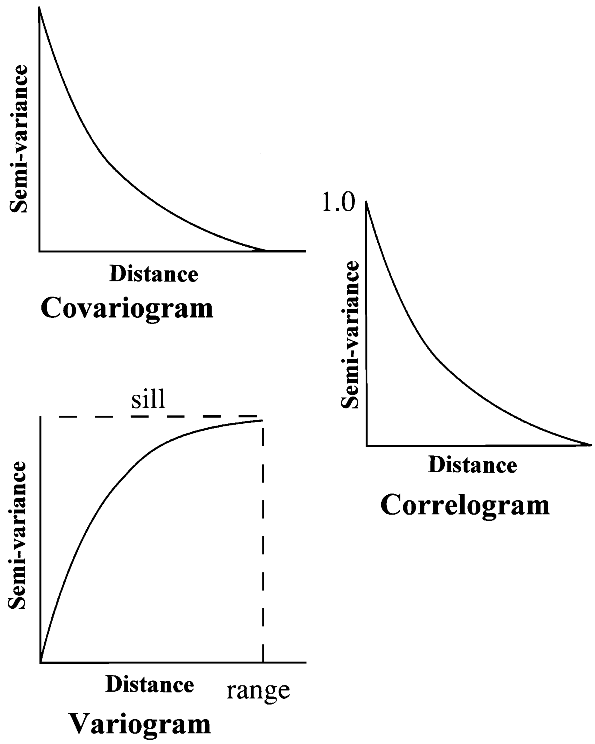
A semivariogram can also be used to determine if spatial patterns are scale dependent. A semivariogram can be defined as:

$$2\hat{\gamma}(d) = \frac{1}{n(d)} \sum_{\text{distance}(i,j)=d} (z_i - z_j)^2$$

where $\hat{\gamma}$ represents the semivariogram, d is the distance between pairs of observations, n is the number of observation, and z_i and z_j are the observed values [4]. In a semivariogram, the variance at zero distance is called a nugget effect (Fig. 1). The distance at which the semivariogram plateaus is known as the range. There is no scale dependence beyond the range and the value becomes constant. This constant value is termed the sill (Fig. 1). The difference between the nugget and sill reflects the proportion of the total variance associated with scale dependency. It has been suggested that the estimates of autocorrelation and correlograms provide a better technique than a semivariogram approach because semivariograms assume no spatial relationship in estimates of the mean and the variance [3,5]. A correlogram is a plot of autocorrelation values against distances between pairs of observations (Fig. 2). Autocorrelation describes the common property of spatial data, which is that a given spatial variable is more likely to be similar at closer locations than at distant locations. In other words, spatial data can often be correlated with itself at short distances [5].



Scale, Effects, Figure 1 A semivariogram [3,4,5]



Scale, Effects, Figure 2 Covariogram, variogram (semivariogram), and correlogram [5]

Fractal dimensions have been suggested as a tool for measuring the effects of scale [2,3,6]. Fractal dimensions are based on the idea that many spatial phenomena are statistically self-similar. In other words, the geometry of an object repeats itself at multiple scales and the geometry of the whole object is similar to the geometry of a portion of itself at a larger scale. Burrough first suggested in 1981 that there is a scale at which the self-similarity changes [3]. Therefore, changes in fractal dimensions with changes in scale can be used to describe the effects of scale. Fractal dimensions of earth objects are now often measured as an index of complexity in their spatial patterns. The fractal dimension, D, can be expressed as:

$$D \approx \frac{\ln(A)}{\ln(L)}$$

where A is the area of an object and L is the perimeter of the object. No change in D indicates self-similarity of the object, while a change in D indicates the scale at which self-similarity no longer holds. When the fractal dimension index is measured at varying scales to study a spatial process, the highest fractal dimension can indicate the scale at which the process of interest operates. Fractal dimensions have been used to characterize topography and terrain, landscape types, coastlines, political boundaries, and urban landscape development. Fractal dimensions can also be used with time-series data since fractal dimensions measure any object that looks the same at varying scales, spatial and temporal.

Another important issue relates to the effects of aggregation and disaggregation methods [6]. Spatial data can be disaggregated using resampling approaches. However, disaggregation of spatial data is often limited by the resolution of the data. In a multilayer dataset, spatial data should not be disaggregated to the scale of the finest resolution available. Instead, spatial data should only be disaggregated to the coarsest sampled resolution used to create the dataset. Spatial data aggregation is less limited and can be accomplished more easily compared to disaggregation. The most commonly used aggregation methods are the average, dominant values, and sampling every Nth cell methods [3]. Each of these methods has different effects on the spatial data analyzed and the associated results. The average method uses a moving window. The moving window has a larger cell size (i. e., a coarser resolution) and it estimates the average value within the finer resolution cells in the original dataset. This method decreases variance and smoothes the data, but increases spatial autocorrelation. The dominant value method also resamples the data using a larger cell and creates a dataset at a coarser resolution. In contrast to the average method, the dominant value method reduces spatial autocorrelation. Sampling every Nth cell is a subsampling approach where every Nth cell is selected from the entire dataset to provide a subset. Similar to the dominant value method, this method decreases spatial autocorrelation. These aggregation methods are more commonly used with raster data than with vector data. Vector data need to be converted to a raster format before these methods can be applied.

Key Applications

Ecology

Effects of scale have attracted a great deal of research attention from ecologists for several decades. Advancements have been made in both theoretical and empirical aspects in many subdisciplines, including aquatic ecology, plant ecology, and wildlife ecology. The hierarchy theory was pioneered in ecology and ecologists often use the framework of hierarchical levels to integrate effects of scale in their work. In hierarchy theory, processes can be observed at one distinct scale or across multiple scales. Rescaling theories have also been well developed and commonly used in ecology. Isometric scaling and allometric scaling approaches are widely used [6]. Isometric rescaling refers to changes in scale using direct proportions. For instance, changes in volume by a factor of 10 can change mass by the same factor. Allometric scaling refers to changes in scale using indirect proportions (i. e., power function). Allometric scaling function can often be expressed as an exponent. Allometric scaling is more commonly used in studies of environmental changes over time. Fractal analysis is also commonly used in studies of land cover types and their changes. Regression-based hierarchical modeling has been proposed for modeling coupled human–environment systems.

Geosciences

Geographers were the first to recognize the effects of scale. Physical geographers and human geographers both deal with spatial phenomena that span across multiple scales. Geologists, geomorphologists, and soil scientists also study earth processes and patterns that are scale dependent. For example, stream channels, shoreline erosion, soil erosion by wind and water, hillslope processes, and sediment accumulation all vary across spatial and temporal scales. Biogeochemical cycles and hydrological processes can be studied at multiple spatiotemporal scales ranging from millimeters to hundreds and thousands of kilometers. Fractal geometry has been applied in describing topography, shoreline patterns, slope and aspect distribution, and human population data including the distribution of cities, population density, income, and jobs. Other methods such as geographically weighted regression have been used in modeling the distribution and sources of diseases.

Social and Political Sciences

Many social and political processes operate across spatial and temporal scales. Understanding social processes requires understanding of social scales. Social scientists argue that scale is often a result of combination of factors including cultural values, social practices, and political and economic forces. In social sciences, the scale theory is based on variables such as state, capital, political parties, political activists, and nonstate level political factors [1]. The organization and reorganization of local, regional, and global economies, their supporting labor forces, and industrial activities are important topics related to the theory of scale in social sciences. Furthermore, social scientists, along with political economic geographers and human geographers, argue that scale is not a function of space [1]. The relationship between space and scale and the organization of scale are now among the hot topics in social science. Issues of scale are also an important topic for political scientists. In particular, aggregation and disaggregation methods and their implications are important for political scientists and politicians. For example, aggregates and disaggregates of political election results at different spatial and temporal scales might reveal different outcomes and patterns. County level, state level, and regional level election results might all reveal different variances and hence potentially different patterns. Indeed, politicians could determine a particular scale at which spatial patterns might favor their arguments.

Economics

Economic processes and patterns are often scale dependent. Underlying economic forces and income sources might vary from one location to another when examined at a given scale. However, the patterns might be uniform at a higher level at coarse scales. Housing prices, for instance, might have a large variance when individual towns are compared. Resort towns might have a much higher average of housing prices, while other towns have lower prices. When prices are aggregated at regional levels, however, this variance might no longer be noticeable.

Future Directions

The term scale has several different meanings. Theoretical and empirical advances are being made within each different meaning. Further development, however, is still necessary. In terms of cartographic scale, researchers are still in the process of deciding which features at what detail should be included in maps of specific scales [1]. In terms of scale as dimensions of observations, scientists are working on how to incorporate scale into the analyses, how to determine operational scales, and how to measure changes in the relationships between variables as scales change. Physical geographers adopt the hierarchy theory in their study of spatial processes and resulting patterns. The hierarchy theory allows them to organize natural phenomena into distinct scales associated with levels and to simplify multiscalar analyses. However, some physical geographers do not readily accept the application of hierarchical theory and argue that different spatial and temporal scales require different models [1]. They suggest that instead of simplified models, we need dynamic models for very small/short spatiotemporal scales and very large/long spatiotemporal scales. Physical geographers also argue that interpretation across scales is an important issue.

Human geographers also approach the concept of scale differently and do not readily accept the hierarchy theory [1]. Rather than modeling fixed sets of levels and the associated variables with their properties, human geographers ask questions of how scale comes into existence, how it is constructed and reconstructed, and how it changes over time. They argue that political reorganization, social reproduction, and capital accumulation change human social scales. In other words, scales in human geography are not always associated with space and spatial units, e. g., boundaries of a county or a country can change as local and regional politics change. The future challenges in human geography, therefore, is in understanding how these changes occur and how causal factors and forces relate to each other and in which direction (i. e., top-down or bottom-up etc.). The ideas of changing scales in dynamic human and environment systems are also becoming more common in other disciplines, so that there is no one fixed scale that remains permanent over time.

Recommended Reading

1. Sheppard, E., McMaster, R.B.: Scale and Geographic Inquiry: Nature, Society, and Method. Blackwell, Oxford (2004)
2. Tate, N.J., Atkinson, P.T.: Modelling Scale in Geographic Information Science. Wiley, Chichester (2001)
3. Quattrochi, D.A., Goodchild, M.F.: Scale in Remote Sensing and GIS. Lewis, Boca Raton, Boston, London, New York, Washington, D.C. (1997)
4. O'Sullivan, D., Unwin, D.J.: Geographic Information Analysis. Wiley, Hoboken, NJ, USA (2003)
5. Bailey, T.C., Gatrell, A.C.: Interactive Spatial Data Analysis. Prentice Hall, Longman, Harlow (1995)
6. Peterson, D.L., Parker, V.T.: Ecological Scale: Theory and Applications. Columbia University Press, New York (1998)

Scale Rendering

► Scale, Effects

Scan, Sequential

► iDistance Techniques

Scan Statistic

► Homeland Security and Spatial Data Mining

Scene Analysis

► Data Acquisition, Automation

Schema

► Geography Markup Language (GML)

Schema Mapping

► Database Schema Integration

Scientific Visualization

► Exploratory Visualization

Screening Design

► Screening Method

Screening Method

XIAOBO ZHOU, HENRY LIN
Department of Crop and Soil Sciences,
The Pennsylvania State University,
University Park, PA, USA

Synonyms

Screening design; Factor screening method

Definition

The screening method is the identification of a few parameters that have the largest influence on the model outputs. This method aims to provide adequate information about the sensitivity of the model to its input, while decreasing the computation cost. It is particularly useful when dealing with models containing tens or hundreds of input factors with a few very influential ones and a majority of noninfluential ones.

Main Text

Sensitivity analysis is performed to evaluate the influence of input factors on model response. The number of factors could be huge but only a small number of them have significant influences on the model output. A screening method is the desirable approach in such a situation with respect to its low computation cost. However, the gain of reduced computation cost is achieved by sacrificing some sensitivity information, because this type of method often only ranks the input factors in the order of importance on the total uncertainties in outputs, but do not quantify how much a given factor is more important than another, i. e., it provides qualitative rather than quantitative sensitivity information. There are several screening methods, among which the one-at-a-time (OAT) processes are the simplest. In these methods, the impact of changing the values of each factor is assessed in turn. Other screening methods include the Cotter's design, the iterated fractional factorial design (IFFD), and Bettonvil's sequential bifurcation design.

Cross References

► Global Sensitivity Analysis
► Local Sensitivity Analysis
► Sensitivity Analysis

SDI (Spatial Data Infrastructure)

► deegree Free Software

SDSS

► Spatial Decision Support System

SDTS

► Spatial Data Transfer Standard (SDTS)

Search, Multi-Criteria

► Skyline Queries

Search, Multi-Dimensional

► Skyline Queries

Secure Past, Present and Future Tree

► Geospatial Authorizations, Efficient Enforcement

Secure Time-Parameterized Tree

► Geospatial Authorizations, Efficient Enforcement

Security Models, Geospatial

Soon Ae Chun[1], Vijay Atluri[2]
[1] College of Staten Island-City University of New York, New York, NY, USA
[2] Department of Management Science & Information Systems, Rutgers University, Newark, NY, USA

Synonyms

Geospatial authorization; Geo-role-based access control; Location-aware access control; Spatially aware role-based access control; Geotemporal role-based access control; Context-aware role-based access control; Environment-sensitive access control; Context-aware dynamic access control

Definition

Geospatial security models refer to formal approaches to control access either to geospatial entities (objects), i. e., data and application with geospatial references, such as maps, satellite images of earth, business data targeted to a certain region, government statistics of different areas, or to any resources by the users (subjects) based on his or her geospatial location. These georeferenced data or applications are referred to as geospatial objects, and georeferenced users are referred to as geospatial subject. Geospatial security models thus consider the geographic location, and geographic features as a primary feature in the access control to the objects. In addition to the location or geographic area of a geospatial object or subject, other context information such as temporal, directional, and speed data that are often associated with the location is also considered in access control. A model of the context or environment of geospatial subjects, such as the user's location, time, movement and interaction-related features are included as an important construct in geospatial security models.

The geographic reference of the geospatial objects has spatial extent (coverage) that maps to geographic area on earth, often captured by earth observing sensors, while the geospatial subject's location is captured by global positioning system (GPS) devices. The geographic reference of the geospatial entities can be either expressed in geographic coordinates, such as longitude or latitude, or in symbolic names, such as city names. Geospatial objects can be represented with raster data (grid cells) or with vector data that uses geometric shapes, such as points, lines, and polygons.

Historical Background

The Federal Geographic Data Committee (FGDC) has established and implemented metadata content standards [1] for sharing and disseminating geospatial data, which typically include information used to identify geographic data, such as description of data, time period of content validity, bounding coordinates covered, keywords, and access and use constraints. Access constraints specify restrictions and legal prerequisites for accessing the data set. These include any access constraints applied to assure the protection of privacy or intellectual property, and any special restrictions or limitations on obtaining the data set. Use constraints are restrictions and legal prerequisites for using the data set after access is granted. These restrictions typically apply to the entire geospatial data set as a whole, and the access instructions in free text format is geared to human users, hence the difficulty for automated enforcement of these use and access policies.

Location-based security needs are highlighted and authentication using GPS and Cyberlocator is discussed in Denning and MacDoran [2]. The geospatial authorization model (GSAM) has been presented in Atluri and Chun to specify authorization rules for accessing geospatial data, using geographic credentials of spatial objects as well as those of the subjects [3,4,5]. It illustrates the access-control mechanism for high-resolution satellite image data based on the coverage location and resolution levels. Belussi et al. and Bertino et al. present an access-control model for vector-based map data and map services [6,7].

Covington et al. uses environment roles in a generalized role-based access control (RBAC) model and presents context-aware security architecture [8,9]. With mobile and pervasive computing gaining in importance, the mobile user's geolocation and other context information is used to dynamically activate the relevant authorized roles, as in Zhang et al. [10]. Bertino et al. introduces GEO-RBAC as an extension of RBAC to model spatial objects, user positions, and geographically bounded roles. Roles are activated based on the position of the user [11]. Ardagna et al. presents the location-based predicates to express the location conditions of a subject, such as position-based, movement-based and interaction-based conditions [12]. Each location condition carries a confidence level to model the impreciseness of the location service.

Scientific Fundamentals

The geospatial information [13] can be represented as a raster image data type or a vector data type. A raster image data type is a set of grid cells consisting of rows and columns, where each cell contains a value that represents the thematic geospatial data, such as land use or wetlands. On the other hand, the vector data type represents geospatial data using geometric shapes, such as points, lines, and polygons. Schools can be represented as points while a water body can be represented as a polygon. The spatial data are represented by the coordinates of a vector geometry or the position of a raster cell.

The following are some of the characteristics of geospatial data:

- The advances in data collection technologies such as earth orbiting satellites, cartographic tools, geographic information systems (GIS) tools, etc., produce enormous amounts of high-quality geospatial data.
- The geospatial data can be visualized easily in a digital map or images using computer systems such as GIS, as well as web-based map applications.
- These geospatial tools allow easy integration of geospatial data based on geocoordinates. Combinations of geospatial data may allow users to gain new information easily.
- Geospatial data is becoming increasingly accessible via simple web-based systems, aiming at increasing popularity among the general public. For example, Terraserver for satellite imagery, the TIGER mapping service for the US Census data, FGDC Clearinghouse, and Google Map services have all contributed easy access, distribution and sharing to geospatial data.

The easy access, visualization, integration and analysis, and readily available tools are beneficial for various application areas for decision making, such as detecting distress crops or identifying locations for natural resources, as well as many other applications in areas such as environmental monitoring, disaster relief, transportation, utility mapping, urban development, real estate evaluation, infrastructure planning, national security. On the other hand, these characteristics of geospatial data can facilitate misuse and pose a threat to the security of national assets and a threat to privacy of individuals. For instance, while high-resolution low-cost satellite imagery, in combination with other geospatial data such as maps, digital line graphs, census data, voter registration, land-ownership data or land-use data, allows users and organizations to enjoy many benefits, the details in the high-resolution images could reveal vital national resources, such as major airports or nuclear power plants, that could be the target of threats and could encourage industrial espionage, terrorism or more cross-border military attacks. These geospatial data may also reveal manmade objects, such as buildings and backyards of properties, often associated with private owners who want to keep it private. Thus, geospatial data of these types require appropriate access control based on the location, in addition to other characteristics.

GeoSpatial Authorization Model: Raster Data Access Control

Recognizing the potential use of geospatial data as a threat to security and privacy, a Geospatial Authorization Model (GSAM) has been proposed [4,5] that allows specification of authorizations on geospatial objects based on their spatial coverage area, time, and resolutions, among other characteristics associated with the data. GSAM supports emerging policies for prohibiting the release of imagery beyond a certain resolution (such as the guidelines provided by the Department of Commerce), notifying when an image crosses an international boundary, or when such a request is made, etc. [14,15], and it is in alignment with the current practice of security policies of restricting high-resolution imagery distribution in a certain region, (e. g., Space Imaging [16] restricts the collection and dissemination of imagery of Israel beyond a certain resolution).

Authorization An authorization rule is represented with a tuple <*gr, go, pr, τ, sign*>, where *gr* is a geotemporal role expression to denote a set of authorized geotemporal roles, *go* is a geospatial object expression to denote a set of authorized geospatial objects, *pr* is a set of privileges that can be performed on authorized objects, τ is a valid authorization period, and *sign* is a positive or negative sign to denote the permitted or prohibited authorization.

Geotemporal Roles Geotemporal roles for subjects are a set of roles with spatial and temporal credentials indicating that each role can be further differentiated according to location and temporal characteristics. In other words, while a user assumes one role (e. g., a doctor) in a traditional RBAC regardless of where the user is, a user may assume different geotemporal roles dynamically depending on the location, time and different situation. (e. g., a doctor in the Hudson County at fire situation). Thus, a geotemporal role is a role in a scene, represented as the pairing of a traditional *role* for subjects as in a RBAC role hierarchy, and a *scene* that can be associated with a set of geospatial and temporal extents.

Each *scene* can be organized as a generic location hierarchy, e. g., the world contains subregions, or it can be organized in a hierarchy in its symbolic domain. For example, an incident domain may have scenes like fire, flood, earthquake, while a shopping domain may have scenes of mall, retail shop, wholesale area, market, etc. Each scene can be instantiated with scene expression such as scene name, or a specific geotemporal extent, such as < *label, spatial-*

Security Models, Geospatial, Figure 1 Orthophoto image data in Newark Bay, New Jersey (1 m resolution)

extent, temporal-extent > where *label* is a canonical names or descriptive (symbolic) scene name, such as "New York City," "mall" or "fire," *spatial-extent* denotes latitude, longitude, height and width of a bounding box covering a geographic area and *temporal-extent* denotes a time period.

A geotemporal role <role, scene> is defined using a set of spatiotemporal relationships between credential attributes associated with the role and its values, including geospatial and temporal attributes and its values of scenes. A geotemporal role expression is defined as a logical combination of these spatiotemporal relationships between attributes and values. The following illustrates the geotemporal role expressions:

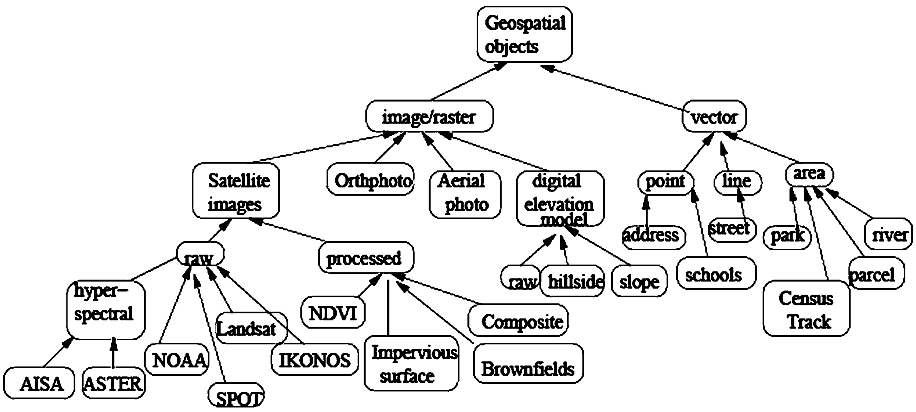
- `<Property_owner AND (home_address equal '123 James Street, Newark, NJ') AND` between`(1990, 2000)>`: denotes all property owners of '123 James Street, Newark, NJ between year 1990 to year 2000.
- `<Bergen-County-police AND` within `(100, 200,10,10) AND` during`(evening)>` denotes all Bergen County policemen who are in the area (100, 200, 10, 10) during the evenings.

Geospatial Objects Geospatial objects include (1) image data such as satellite images, aerial photographs, and scanned data covering the earth's surface (e. g. see Fig. 1), and (2) map data consisting of points, lines, and areas that form shapes and locations of map features such as buildings, streets, or cities (e. g. see Fig. 2).

Tabular data linked to a map feature location or a shape is considered as the description of the map data, e. g., demographic information linked to the map of customer locations. In the case of image objects, the image pixel contents, such as vegetation index (NDVI) or water index or other image contents along with the header information are linked to the images. The geospatial objects are organized in a type of hierarchy as shown in Fig. 3.

Each geospatial object is represented as a set of metadata, a unique *identifier*, the *type* of the geospatial object in the type hierarch, the *latitude, longitude, height* and *width* that represent the minimum bounding box for the spatial region covered by the object, the *resolution*, the *timestamp* (either image download time or last update time), and the *thematic link* to the data set associated with the geospatial object.

Security Models, Geospatial, Figure 2 Vector data with streets, gas stations and zoning types for Newark, NJ



Security Models, Geospatial, Figure 3 Geospital object type hierarchy

Given a geospatial object, a set of access functions {*type; rectangle; resolution; timestamp; data*} return the geospatial object type from the object type hierarchy as shown in Fig. 3, the spatial region covered by the object, the resolution, the timestamp, and the data set linked to the object, respectively.

Geospatial objects can be described with any combination of these access functions and their values. Thus, to denote a set of geospatial objects, *geospatial object expression* is defined by combining a set of spatial characteristics, using these access functions with logical operators such as {>, <, =, >=, <=}, and a set of geometric operators such as {*contain, equal, overlap, meet*} and a set of temporal operators such as {*before, after, during, within*}.

The following shows valid geospatial object expressions:

- (*type = image AND rectangle contains (10,20,10,10) AND timestamp within [Feb2:2002, Mar1:2002] AND resolution > 10m*) denotes a set of images whose spatial extent contains the area (10, 20, 10, 10) downloaded between February 2, 2002 and March 1, 2002, and whose resolutions are higher than 10 m.
- (type = parcel *AND* rectangle *overlap* '123 UniversityAve Newark, NJ' *AND* data = 'propertyownership') denotes a set of parcel maps with links to the property ownership data and with spatial extent overlapping the minimum bounding box that corresponds to the address "123 University Ave, Newark, NJ."
- *Id = LANDSAT26* specifies an image with identifier LANDSAT26.

Geospatial Privileges The privilege modes are essentially of three types: *viewing, copying* and *maintenance*.

The viewing modes include *view*, *view-thumbnail, view-annotation, zoom-in, overlay, identify, animate* and *fly-by*. These operations retrieve data from the data sources and deliver them with basic postprocessing operations, such as *crop* and *mosaicking*, or *geoobject integration* by building specific modules (in the cases of *animate* and *fly-by*). *Zoom-in* allows a user to view an image covering a certain geographic area at a higher resolution; *overlay* allows users to generate composite images, where a composite image is constructed from multiple images by first georegistering and then overlaying them one on top of another (see Fig. 4 for an example); *identify* allows the user to view the tabular data linked to an image; *animate* allows a user to obtain a time series of images and integrate them to show the changes in the images; and *fly-by* allows a user to traverse from one location to another a multiresolution browsing from low-resolution images to high-resolution images.

The copying modes include *download* and *download-data*, which allow source files to be downloaded. Unlike the text data where the display privilege implies the copying privilege, the viewing and copying are distinguished as separate privileges with geospatial data since the objects displayed on the web browser often are image gif files, but not the original source files. The maintenance modes include *insert*, *delete, update* and *compose*. The users with *compose* privilege can create and insert value-added images, using images in the database.

Access-Control Evaluation Given a user request, consisting of <user credentials, area of interest and access privilege>, and a set of authorization rules, the access-control engine evaluates the user's credentials against geotemporal role in each authorization. Once the geotemporal role is activated, the requested object, i. e., *area of interest with certain characteristics such as a specific resolution and timestamp*, and access privilege are evaluated against authorized geospatial object expressions and privileges in each of the activated authorizations. Spatial operations such as {*overlap, contains, equal*} are used in the authorization evaluation. The authorized objects are retrieved and postprocessed to deliver only the authorized area in the geospatial object. The postprocessing may require *cropping* of an authorized area from the geospatial

Security Models, Geospatial, Figure 4
Satellite image and city boundary vector overlay with annotation

object, or *tiling* (i. e., *mosaicking*) of several cropped areas from multiple authorized geospatial objects.

GEO-RBAC: Vector Data Access Control

In [7] and the subsequent work [11], access-control models for vector data were proposed, called the GEO-RBAC model, an extension of RBAC with spatial roles, objects and contextual information. In [7], the focus is on an access control to web map services. The authorization rule is specified with $<r, fc, p, w>$, where the subject *r* is a conventional role in a role hierarchy, object *fc* is a vector object represented as feature classes such as thematic features like roads, towns, etc, privilege *p* is a Open Geospatial Consortium (OGC)-compliant web map services [1,17], and window *w* is the geographic scope of the authorization represented as a polygon. An authorization on privilege is granted if there is an overlap between *feature class* and *the window* for an authorized role. They recognize several web map service-related privileges: the *Notify* privilege controls the execution of the operations for feature insertion and deletion, the *Analysis* privilege controls the execution of the different querying operation, the *ViewGeometry* privilege controls the single operation of *GetFeatures*, and finally *the ViewAttribute* privilege controls the operation of *GetFeatureInfo*.

In [11], the access-control model extends to include the spatial roles to consider the user's actual position that determines the role activation and evaluates the permissions assigned to the roles. The following summarizes the following concepts used in the GEO-RBAC model.

- Spatial objects: the spatial vector data (Hudson River, New York City, etc.) is represented with features that can be mapped onto locations. The location of a feature is represented through *GEO*, that is, a set of geometries, such as points, lines and polygons, or any combination of these types. Different geometries are related by a set of topological relations such as {*Disjoint, Touch, In, Contains, Equal, Cross, Overlap*}. Location of a feature is a geometry in *GEO* type. Features are organized with feature-type containment relationships. For instance, the geometry associated with Town feature type is contained in the geometry of Region feature type. A specific vector object is represented with a minimum bounding box of a feature type, called feature type extension.
- Spatial roles: a spatial role in GEO-RBAC represents a geographically bounded organizational role. The boundary is defined as a feature, such as a road, a city or a hospital. The boundary specifies the spatial extent in which the user is to be located for being enabled to play such a role. The spatial role is defined as a pair $<r, e>$, where *r* is a role name and *e* is the spatial extent of the role, which can be any feature type extension. Thus a doctor role can be associated with different spatial roles to represent doctors of different hospitals' spatial extents.

- Positions: the position of the user is represented with real position (*RPOS*), that is, a geometry type *GEO*, but it is also represented with logical position (*LPOS*) using feature-type extension. A position mapping function maps the real position to the logical position.

GEO-RBAC Model Given these basic constructs, the GEO-RBAC model consists of role *schemas* and *instances, permissions, users* and *sessions*.

Role schema. A role schema is represented as a role name, a spatial extent of role, the logical position feature type, a mapping function that maps the real position to the logical position feature type. A role instance is the role and a specific spatial extent of the role that is a feature type. An example <TaxiDriver, Road-Network, PointOnRoad, m(PointOnRoad) > shows a role schema for a role of a taxi driver on the road network, the position of the role is on PointOnRoad and the real position can be mapped to logical location feature type using mapping function m. Thus a taxi driver at Milan's road can be an instance of the role schema of the example.

Permissions. These are operations performed on spatial objects, such as *Get traffic information* over *Urban-Road-Nework* features, *notify* over *accident features*, or *Find* operation over *Monument features*. There are functions that assign roles schemas to permission sets and functions that map role instances to a set of permission operations.

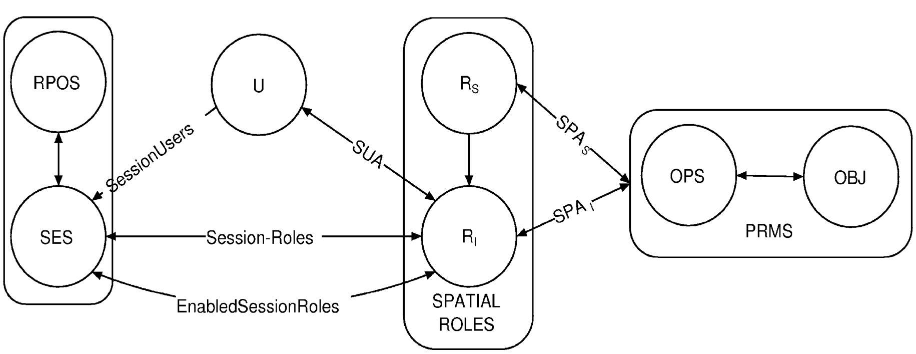
Authorization. Thus the authorizations, as shown in Fig. 5, are specified by the list of permissions created by permission assignment functions, a set of role schemas and role instances, users, sessions, and real positions, and their functions relating these entities. GEO-RBAC uses spatial role hierarchy with partial order relationships among roles. Thus the partial order relation $r1 \leq r2$ denotes that the spatial extent of *r2* is contained in the spatial extent of *r1* and all the permissions allowed for *r1* can be inherited to *r2*. An instance of *r2* then inherits all the permissions of *r2* and those of *r2*. For example, the following shows parts of the GEO-RBAC authorization model:

- Citizen in a city: <citizen, city, pointonRoad, m(pointonRoad)>
- Taxi driver on the road: <taxiDriver, urbanRoadNetwork, poinonRoad, m(pointonRoad).
- Permission 1: (GetTrafficInfo, Ext(urbanRoadNetwork).
- Permission 2: (Notify, Ext (Accident)).
- Assign(citizen in a city) $= \{$*Permission 1, Permission 2*$\}$, is an authorization for a citizen in a city to be able to access traffic information on the urban road, and notify the accidents.
- Assign (taxi driver on the road) $= \{$*Permission 1, Permission 2*$\}$ is an authorization for a taxi driver on a road network can access traffic information on the road, and notify accidents.

Access Request and Access Control The *users* are assigned to role instances, using a spatial role-to-user assignment function. When a user logs on and a session is created, the authorized roles should be enabled in the session. This is achieved by computing the logical position from the user's real position using the mapping function in the role schema. If the location position computed by the location mapping function is contained in the spatial extent of the logical position specified in the role schema, then the role is activated. Thus when an access request is made using <*Session, Real position, operation, object*>, the logical position of the user is used to enable the roles, and determine if the permissions assigned to the enabled roles contains the requested operations on the objects.

Location-Based Access Control

Location-based access control (LBAC) techniques consider user's physical location to determine their access privileges, thus focusing on how to specify user's location conditions in access-control policies. In [12], the access-



Security Models, Geospatial, Figure 5

Type	Predicate	Description
Position	Inarea (user, area)	Evaluate whether user is located within area
	Disjoint(user, area)	Evaluate whether user is outside area
	Distance(user, entity, min, max)	Evaluate whether the distance between user and entity is within the interval between minimum and maximum distance
Movement	Velocity(user, min, max)	Evaluate whether user's speed falls within the range between minimum and maximum
Interaction	Density(area, min,max)	Evaluatewhether the number of users currently in area falls within the interval between minimum and maximum
	Local density(user, area, min, max)	Evaluate if the density is within a "relative" area surrounding the user

Security Models, Geospatial, Table 1 Three types of location-based predicates (reproduced from [12])

Security Models, Geospatial, Table 2 An example of extended truth table for location predicates

Predicate	Confidence thresholds		Maximum tries
	Lower	Upper	
Inarea	0.1	0.9	10
Distance	0.2	0.8	5
Density	0.3	0.7	3

control rules are defined as <*subject expression, object expression, action*>. In order to specify the subject with location conditions and the role-related generic conditions, they define three types of location-based predicates: position-based predicates, movement-based predicates and interaction-based predicates. Table 1 summarized these predicates.

These predicates allow specification of LBAC policies such as "System administrators are authorized to configure the mobile network if they are in the server farm area, they are alone in such an area, and move at walking speed at most."

The determination of the user's actual location to evaluate the location conditions in the access rule depends on the location service, which is sometimes unreliable and can be timed out. To consider this, in [12] confidence and timeout features were used to determine whether the location service on the location predicates are valid. They introduced an extended truth table (ETT) for evaluating location predicates based on confidence thresholds, as illustrated in Table 2.

If the confidence level for a given predicate is greater than the preset upper bound threshold, then the value of a predicate evaluation from the location service is confirmed, but if it is lower than the lower threshold, the negation of the predicate evaluation is confirmed. However, if the returned value is between the upper and lower bound, then the result is neither confirmed nor negated, triggering re-evaluation of the predicates. Similarly, the predicate evaluation iterates until maximum tries and then times out with undefined value.

Access-control evaluations are performed first on the generic predicates, then, if needed, evaluate the location predicates to avoid the costs associated with the use of the location service, and finally determine the truth value based on these two types of predicates.

Key Applications

National Security

High-resolution images can be used for surveillance and intelligence analysis of critical national assets. The location-based control can be enforced by releasing low-resolution images and limiting access to high-resolution images. An intelligence agent at a certain location should be able to gain access to images and other data of a certain region for timely and accurate assessment of the location, while the public or known and potential terrorist groups have limited access to the geospatial data. Baker et al. reports the potential risks on homeland security from publicly available geospatial data that can be misused by terrorists and other adversaries for attacking critical national infrastructure (e. g., agriculture, water, food, public health, telecommunications, transportation, banking and finance, energy, chemical industry and hazardous materials, postal and shipping, governments, etc.) and key national assets (e. g., locations of cultural significance, special event location, military installations, etc.) [18].

Military

A geospatial access control is needed in the military to limit access to critical military resources based on unauthorized individuals' locations or the location of resources. For example, only authorized users at a certain location (e. g., generals in Guam) or facility (a nuclear power plant) are allowed to control launching a missile only when the missile is positioned in a certain location.

Health Care

There are many medical services differentiated by patients' medical conditions and insurance coverage, and doctors' affiliations. In addition, medical providers, such as doctors or nurses, are equipped with mobile devices, providing services at a certain location. Some medical services may be available only to nurses in a certain location, e. g., in intensive units of New York City hospitals, and others are only accessible by doctors in New Jersey hospitals.

Privacy Applications

Mobile users may not want to reveal their location information. Revelation of personal location may expose them to unwanted target marketing, or can cause denial of insurance services in the case of one's location in risk areas. In addition, many governmental geospatial data contains names, property ownership information, and other identifying information and these can be linked to high-resolution imagery. The proposed access-control models for geospatial data and locations can be applied in the privacy protection mechanisms.

Future Directions

The current authorization verification engines support the precise specification of authorizations, but there is a need to address fuzzy requests and fuzzy authorization specifications to address partial credentials and fuzzy locations, especially where technologies fall short of the precise measure of user location. The privacy issue may be partially supported with the geospatial access-control models based on the user locations, but user's trajectory of preferred locations or patterns identified from mining tools can still threaten privacy [19]. Specifying access-control rules manually can be semiautomated with data mining tools. Emerging trends will mine the geospatial roles, and policies from the usage of access patterns and textual specification of policies for constructing actionable access-control rules. With ad-hoc networks assembling and dissembling based on the location, peer-to-peer authentication and privacy issues will need to be addressed.

Cross References

► Geospatial Authorizations, Efficient Enforcement
► Metadata

Recommended Reading

1. Federal Geographic Data Committee (1998) Geospatial Metadata Standards. http://www.fgdc.gov/metadata/geospatial-metadata-standards. Accessed 23 Mar 2007
2. Denning, D.E., MacDoran, P.F.: Location-based authentication: grounding cyberspace for better security. Computer Fraud and Security Bulletin, February 1996, pp. 12–16 (1996)
3. Chun, S.A., Atluri, V.: Protecting privacy from continuous high-resolution satellite surveillance. In: Proceedings of the 14th IFIP 11.3 Working Conference on Database Security (IFIP WG11.3 2000), Scoorl, The Netherlands, August 21–23, 2000, pp. 399–420 (2000)
4. Atluri, V., Chun, S.A.: An authorization model for geospatial data. IEEE Trans. Dependable Secure Comput. **1**, 238–254 (2004)
5. Atluri, V., Chun, S.A.: A geotemporal role-based authorization system. Int. J. Inf. Comput. Security (in press, 2007)
6. Belussi, A., Bertino, E., Catania, B., Damiani, M.L., Nucita, A.: An authorization model for geographical maps. In: Proceedings of the12th ACM International Workshop on Geographic Information Systems, ACM-GIS, November 12–13, 2004, Washington, DC, USA, 2004
7. Bertino, E., Damiani, M.L., Momini, D.: An access control system for a web map management service. In: Proceedings of the 14th International Workshop on Research Issues on Data Engineering, RIDE '04, Boston, USA, March 28–29, 2004
8. Covington, M. J., Long, W., Srinivasan, S., Dev, A. K., Ahamad, M., Abowd, G. D.: Securing context-aware applications using environment roles. In: Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies. SACMAT '01, May 3–4, 2001, Litton-TASC, Chantilly, Virginia, USA 2001
9. Covington, H.J., Fogla, P., Zhan, Z., Ahamad, M.: A context-aware security architecture for emerging applications. In: Proceedings of the18th Annual Computer Security Applications Conference (ACSAC '02), December 10–14, 2007, Miami Beach, Florida 2002
10. Zhang, G., Parashar, M.: Dynamic context-aware access control for grid applications In: Proceedings of GRID, Phoenix, Arizona, November 17, 2003
11. Bertino, E., Catania, B., Damiani, M.L., Perlasca, P.: GEO-RBAC: a spatially aware RBAC. In: Proceedings of the Tenth Symposium of ACM Access Control Technologies, SACMAT '05, Scandic Hasselbacken, Stockholm, June 1–3, 2005
12. Ardagna, C.A., Cremonini, M., Damiani, E., De Capitani di Vimercati, S., Samarati, P.: Supporting location-based conditions in access control policies. In: Proceedings of ASIACCS'06, March 21–24, 2006, Taipei, Taiwan
13. Bossler, J.D., Jensen, J.R., McMaster, R.B., Rizos, C. eds.: Manual of Geospatial Science and Technology. London: Taylor and Francis, 2002
14. Baker, J.C., O'Connell, K.M., Willamson, R.A. eds:. Commercial Observation Satellites - At the Leading Edge of Global Transparency, Rand Corporation:Arlington, VA (2001)
15. Wright, R.: Private eyes. New York Times Mag. September 5, 1999, Final Section 6, Page 50, Column 1
16. Space Imaging Middle East (2004) http://www.spaceimagingme.com/content/faq/Questions/index.asp. Accessed 23 March 2007
17. OpenGIS Consortium (2006) OpenGIS Web Map Service Implementation Specification.http://www.opengeospatial.org/standards/wms 23 Mar 2007
18. Baker, J.C., Lachman, B.E., Frelinger, D.R., O'Connell, K., Hou, A., Tseng, M., Orletzky, D., Yost, C.: Mapping the Risks: Assessing the Homeland Security Implications of Publicly Available Geospatial Information. Rand Corporation, Santa Monica, CA (2004)

19. Ye, N.: Handbook of Data Mining. Laurence Earlbaum, Mahwah NJ (2003)

Selected Pattern

► Frequent Pattern

Selection Criteria

► Co-location Patterns, Interestingness Measures

Self Organizing Map (SOM) Usage in LULC Classification

SUDHANSHU SEKHAR PANDA
Department of Science, Engineering, and Technology, Gainesville State College, Gainesville, GA, USA

Synonyms

Kohonen map; Self organizing map usage; SOM usage; Artificial neural network

Definition

One of the unsupervised clustering techniques that is widely used for data dimensionality reduction with topological preservation is Kohonen's self-organizing map (SOM). It is a subtype of artificial neural networks. Therefore, SOM is used for visualizing low-dimensional views of high-dimensional data such as classification or grouping. SOM networks are based on competitive learning, i. e., the "winner takes all" approach [1]. In this process, the individuality of the data is rarely lost; rather it is preserved within the winning output neurons of the clusters. It is based on human brain and sensory input [1,2]. This characteristical approach of the system (SOM) makes it superior or at best competitive to other unsupervised classification techniques used in image classification, data reduction, or clustering mechanism.

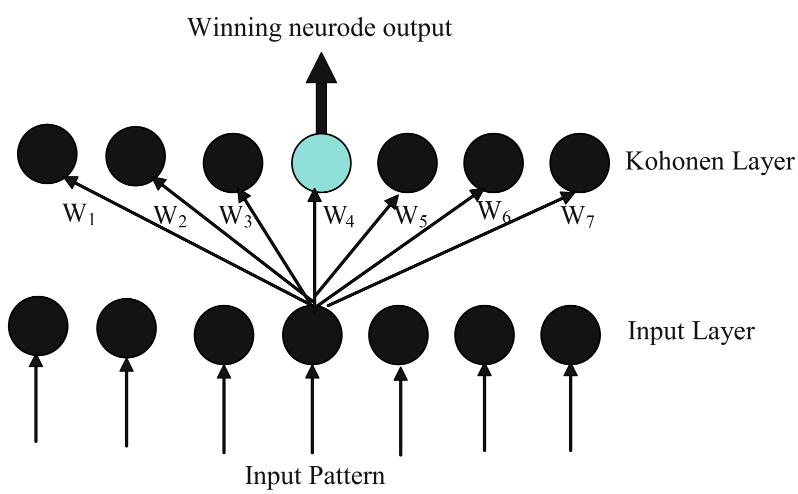
Historical Background

Unlike most clustering techniques, the SOM constructs a topology preserving mapping from the high-dimensional space onto map units in a way that relative distances between data points are preserved. The map units, or neurons, usually form a two-dimensional regular lattice where the location of a map unit carries semantic information. The similarity in input patterns is preserved in the output space in the process of data compression and dimensionality reduction [1,3]. Because of its typical two-dimensional shape, it is also easy to visualize. The model was first introduced by the Finnish professor Teuvo Kohonen in 1982. With reference to its developer, the SOM is also known as the Kohonen map. The architecture of a typical Kohonen one-dimensional SOM network is shown in Fig. 1.

Scientific Fundamentals

Network Structure

The development of the SOM is synonymous to the human brain. The brain is organized with sensory inputs represented by topologically ordered computational maps. Sensory inputs such as tactile [4], visual [5], and acoustic are mapped onto different areas of cerebral cortex in a topologically ordered fashion. As Haykins [1] describes it, "Computational maps in the brain is the principle of topographic map formation, which may be stated as Kohonen [6]: The



Self Organizing Map (SOM) Usage in LULC Classification, Figure 1 Architecture of a typical Kohonen one-dimensional SOM network [2] spatial location of an output neuron in a topographic map corresponds to a particular domain or feature of data drawn from the input space." As shown in Fig. 1, the SOM is a single layer feedforward network where the output syntaxes are arranged in a low dimensional (usually 2D or 3D) grid or topology. Each input is connected to all output as shown in the SOM architecture in the Fig. 1. A random weight vector ($W_1,\ldots,W_n$, as shown in the Fig. 1) with the same dimensionality as the input vectors are attached to each neuron. The number of input dimensions is usually a lot higher than the output grid dimension because the output neurons are the culmination of winning neurons only. SOMs are mainly used for dimensionality reduction rather than expansion. The working procedure is further explained in the following section.

General Working Procedure of Kohonen's SOM

Kohonen's [7] SOM consists of two layers: an input layer and a competitive layer known as the Kohonen layer, or SOM layer [2,8] (Fig. 1). These two layers are fully connected. Each input layer neuron has a feed forward connection to each neuron in the Kohonen layer [2].

A weight vector (w) is associated with each connection from the input layer to a neural unit (Kohonen layer). Inputs (I_j) to the Kohonen layer are calculated with the following equation:

$$I_j = \sum_{i=1}^{n} w_{ij} x_i \,, \tag{1}$$

where w is the weight vector, x is the input vectors, and w_{ij} x_i is a dot product. As the SOM technique works with the approach of winner takes all, thus, the output winning neuron becomes the neuron with the biggest I_j. The winning neuron is chosen by finding the neuron that has a weight vector with the minimum Euclidean distance, d_j, from the input vector [2], i. e.,

$$d_j = \|w_i - x_i\| \,. \tag{2}$$

In the case of the Euclidean distance calculation, the weight, w, and the input vector, x, are not normalized. However when training the network, one conscience mechanism is used which requires some additional steps. The conscience mechanism adjusts the distances to encourage non-winning processing elements (PEs) to win. The adjusted distance, d'_j, can be calculated by the following equation:

$$d'_j = d_j - B_i \,, \tag{3}$$

where B_i is the bias factor that can be calculated by

$$B_i = \gamma \left[\frac{1}{N} - F_i\right] , \tag{4}$$

where F_i is the frequency with which the PE i has historically won. At the initialization stage, the F_i is $1/N$ (inverse of the number of observations), and B_i is thus zero at initialization.

A beta (β) factor keeps track of the winning frequency of each PE. The following equations are used to calculate F_i (used in Eq. 4) for the winning PE (with the lowest adjusted distance) and all other Kohonen PEs, respectively,

$$F_{i(\text{new})} = F_{i(\text{old})} + \beta\left(1.0 - F_{i(\text{old})}\right) , \tag{5a}$$

$$F_{i(\text{new})} = F_{i(\text{old})} + \beta\left(0.0 - F_{i(\text{old})}\right) . \tag{5b}$$

The β is considered as the inverse to the number of observations used for classification. The SOM learning iterations are followed up to a maximum of 30 times the number of observations, which are thereby used for the learning stoppage iterations [9].

The Kohonen net is trained by competitive learning. The neurons in the Kohonen layer compete with each other to be the winning one when the input neuron is added to the layer. The winning neuron is always trained with the following equation:

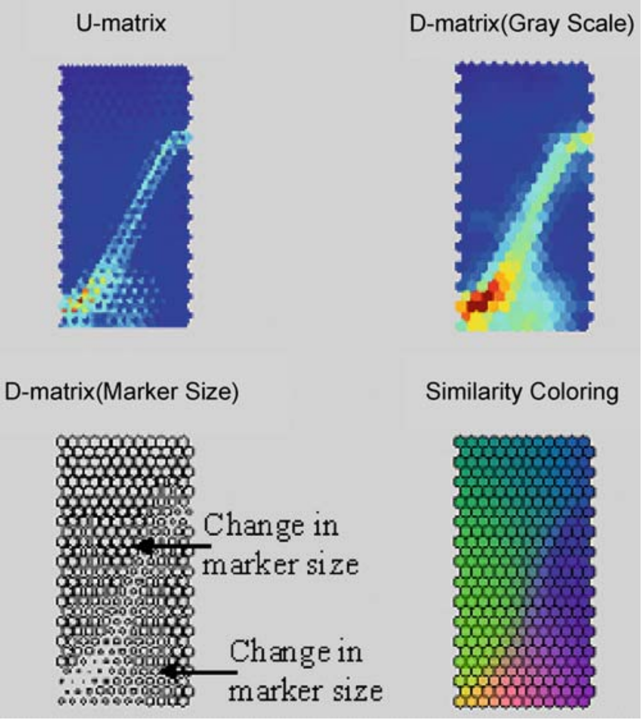
$$W_{ij}^{\text{new}} = W_{ij}^{\text{old}} + \eta\left(x_i - W_{ij}^{\text{old}}\right) \tag{6}$$

where η is the learning parameter. The weight vectors corresponding to the winner and the neural units in its topological neighborhood are updated to align them towards the input vector. The learning rate is updated from time to time, and the neighborhood size is reduced during the course of the learning process [10]. The SOM then determines the output cluster (expected) values of the input vector.

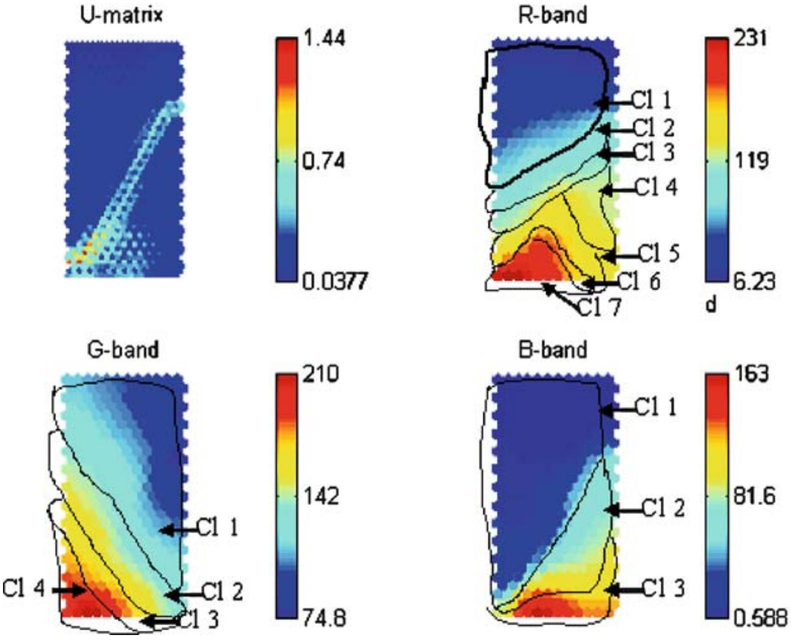
S

Examples of Kohonen's SOM Application in Image Classification

Visualizing the SOM: The SOM is easy to visualize. Several visualization techniques have been devised to visualize the SOM clustering process. These visualization processes are important to determine the number of distinct clusters in an image. Typical U-matrix, D-matrix (gray-scale), and D-matrix (marker size) representations of image data show the cluster numbers in each image (Fig. 2). SOM Toolbox is a free software tool available for download from the internet [11] which implements the Self-organizing Map (SOM) algorithm as a function package for MATLAB 5 (Math Works, Inc., Natick, MA). The SOM Toolbox may be used to perform clustering analysis in images in order to identify the number of distinct clusters present. The U-matrix is known as a unified matrix that shows the general cluster form of the data points derived

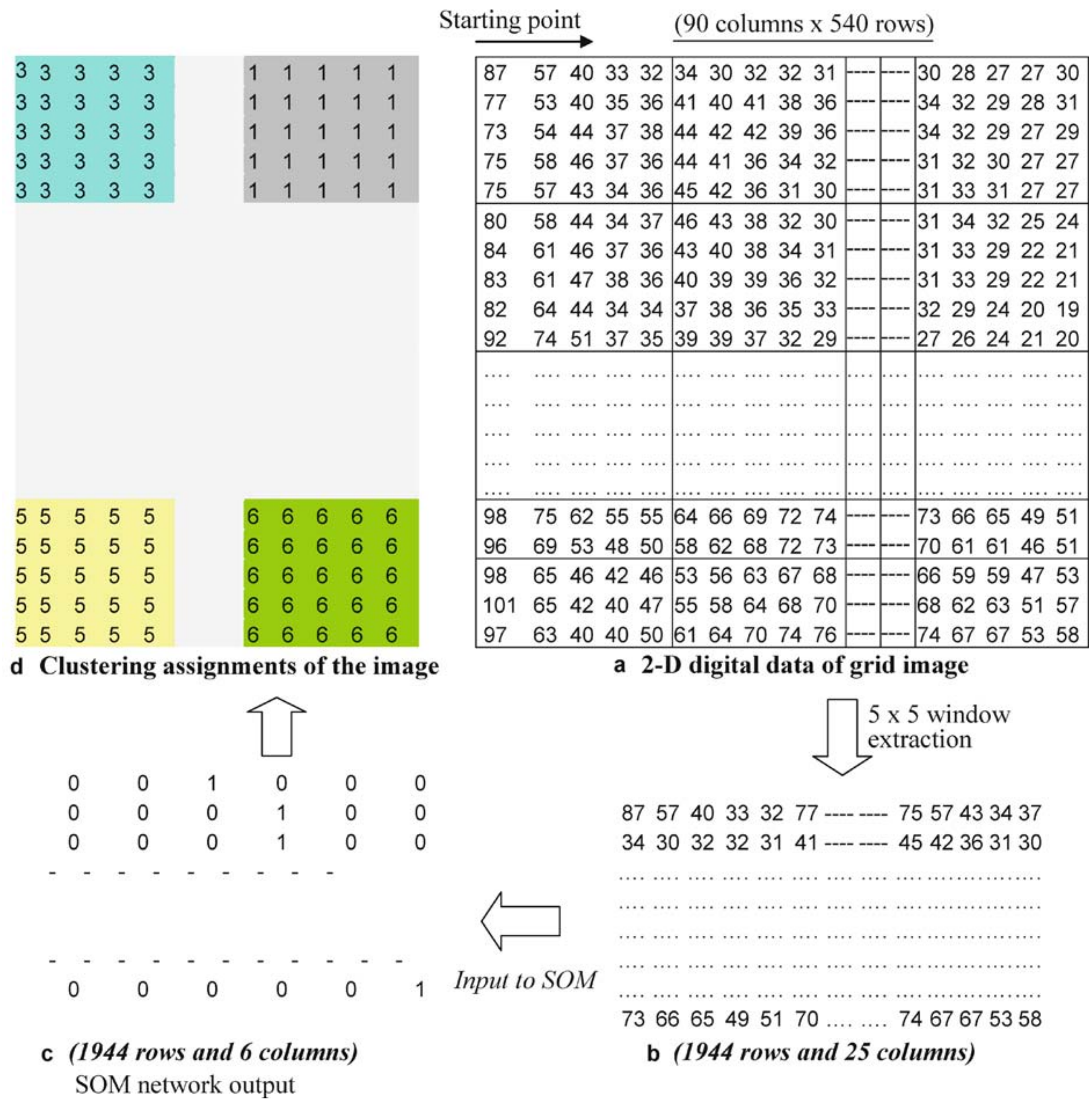


Self Organizing Map (SOM) Usage in LULC Classification, Figure 2 Typical U-matrix, D-matrix (*gray-scale*), and D-matrix (*marker size*) representation of image data with SOM Toolbox analysis



Self Organizing Map (SOM) Usage in LULC Classification, Figure 3 Typical individual image band matrices of the composite band image data with SOM Toolbox analysis [12]

from the image. In the U-matrix graph, each change in color and hexagon size represents a cluster group at either side of it. Again, the same color and size of hexagons establish the groups as the same cluster. U-matrix is a partition matrix that contains the partition labels of different gray values belonging to similar groups. D-matrix is the data matrix obtained after SOM clustering. In the D-matrix cluster map, a change in the marker size from one form to another suggests the presence of two separate clusters. Fig. 3 visually describes the number of clusters present in several images tested using the tools mentioned earlier. The clusters are graphically shown in the images.



Self Organizing Map (SOM) Usage in LULC Classification, Figure 4 Data arrangement and SOM classification for clustering of a typical aerial image

Image Clustering Using SOM: SOM unsupervised clustering of images can be conducted using several commercial softwares. One of the software used for the purpose of SOM unsupervised classification has the ability to create the initial architecture SOM network. Generally, for image classification, a two dimension SOM is created with only two layers, i. e., input layer and Kohonen layer. The user has the option to chose the winning neurons from a set of input blocks comprised of non-overlapping windows of 2×2, 3×3, $5 \times 5, \ldots$ pixels (Fig 4a). That means a winner will be chosen by a SOM algorithm from that window of pixels, such as four pixels from the 2 x 2 window or 9 pixels from the 3×3 windows. The 2-D images are scanned from left to right and in a top to bottom fashion as shown in Fig. 4b. Image gray values may be normalized to make them compatible for SOM processing using the following relationship:

$$X_n = \frac{X_i - \mu_x}{\delta_x}, \tag{7}$$

where X_n is the normalized value, X_i is the ith input variable, μ_x is the mean of all input variables, and δ_x is the corresponding standard deviation. A program in Visual C++ or any other programming languages can be developed/written to arrange the image digital values (ASCII data) into the data arrangement format shown in Fig. 4a and the data normalization if required. Then the dataset need to be fed to other commercial software designed for the SOM clustering. An expected number of classes from the image is usually provided by the user as input. Thus, the SOM algorithm of the "winner takes all" approach chooses the winning neuron (pixel gray value for images) out of the group of pixels in that window and the gray value of the winning pixel is related to a class (say 1 to 10, as provided by the user). As shown in Fig 4c, a value of '1' is assigned to the class for that group (2×2, $3 \times 3, \ldots$ window) and others in the group get values of '0'. Finally, all the pixels of that group take the winning pixel's assigned class (by a SOM algorithm) (Fig. 4d). Thus, the entire image is classified into different cluster groups.

A square or circular neighborhood may be used along with the Euclidean distance, Gaussian distance, or other distance measures for selecting the winner from each input (corresponding to each window of the grid image). A bias factor (β) may be used as the inverse of the number of observations/rows used in the network. A result of a typical aerial image classification using the SOM unsupervised classification technique is shown in Fig. 5b along side the original image (Fig. 5a).

Key Applications

Business

In business, the Traveling Salesman Problem *(TSP)* is a complicated maze to navigate. It consists of finding the shortest (closed!) path connecting a number of points (e. g., cities). The SOM is used to adjust itself to the given inputs (cities), creating a path close enough to the perfect one. Fig. 6a and 6b provide a visual representation of the solved problem (courtesy of www.e-nuts.net). The cities (in a 3D-environment) and the network neurons are shown as points, and the links between neurons are lines.

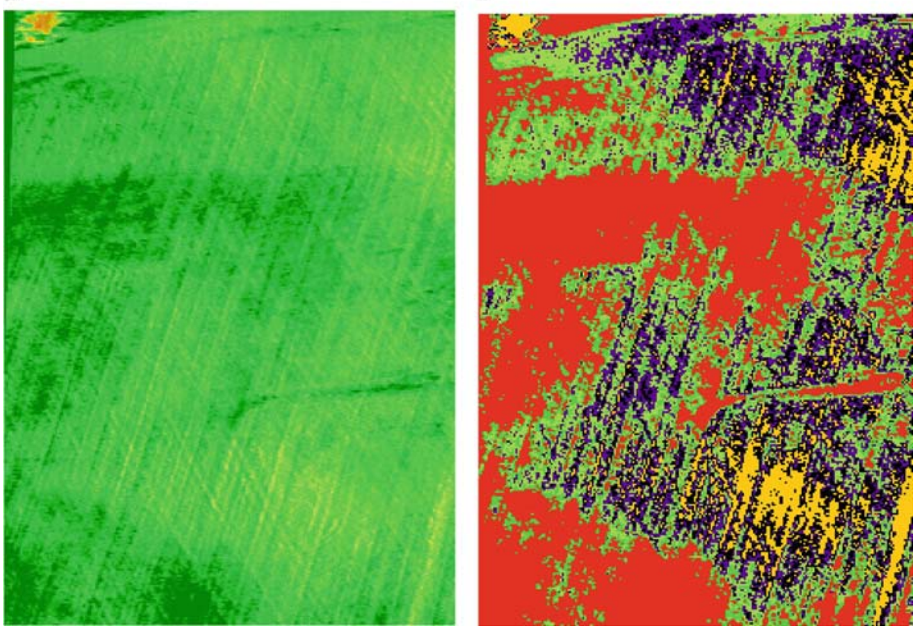
Language Processing

Kohonen's Self-Organizing Map (SOM) can be used to make word category maps (WCM). WCMs are SOMs that have been organized according to word similarities. Conceptually, interrelated words tend to fall into the same or neighboring map nodes, i. e., word categories. With no a priori information, the SOM generates a model of the word classes [13]. This process could be of great help to libraries.

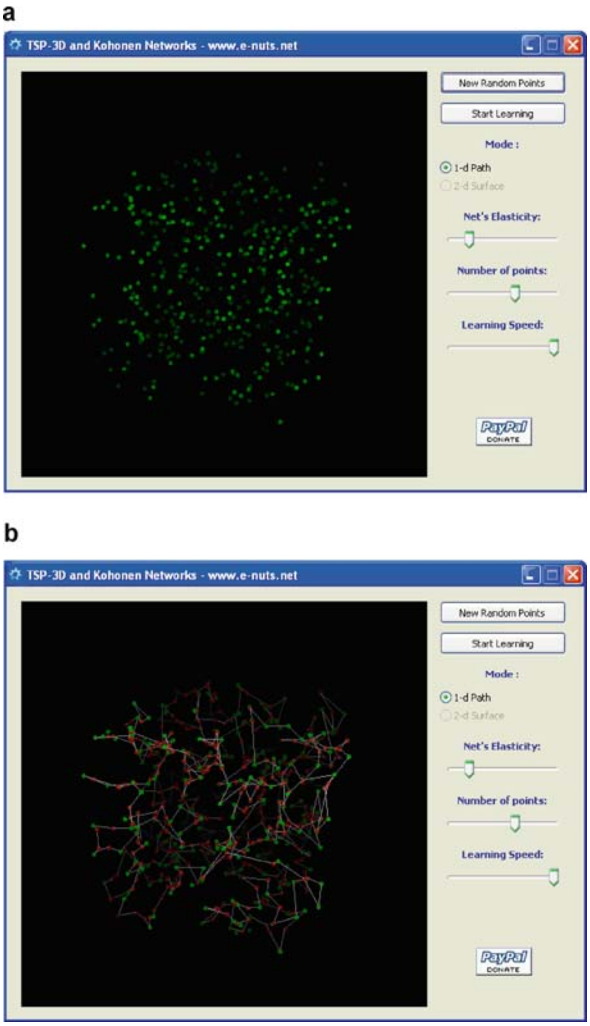
Pattern Recognition

A Kohonen neural network is an iterative technique used to map multivariate data. Additionally, they are very useful for data mining. Data preprocessing for SOM applications is usually minimal and data outliers usually do not affect the SOM classification or pattern recognition as outliers

a b



Self Organizing Map (SOM) Usage in LULC Classification, Figure 5 **a** Typical False color composite (*FCC*) image. **b** Classified image using SOM unsupervised clustering technique



Self Organizing Map (SOM) Usage in LULC Classification, Figure 6
a 3-D representation of cities as points **b** shortest path determination using SOM

only affect one map unit and its neighborhood, and not the entire process.

Speech Analysis

The clustering algorithm of the SOM is used extensively in speech analysis in areas such as voice matching. Its multi-dimensional data mapping is an advantage in speech analysis.

Robotics

Many tasks in robotics are difficult to tackle with explicit models. Neural networks with their inherent learning ability offer feasible alternatives to more traditional approaches. Self-Organizing Map NN techniques contribute to the solution of robotic tasks, such as map building, object recognition, and the coordination of multi-joint movements.

Industrial and Medical Diagnostics

Medical or industrial diagnostics includes image analysis. As this article discusses SOM applications in image clustering and classification, they are widely used in medical image analysis and also in industrial fault scanning. Since medical images are smaller in size, the SOM can be used with high efficiency.

Instrumentation and Control

Fault tolerance study in instrumentation is being conducted with the use of the SOM. The future of the SOM lies is in this area.
There are hundreds of other fields where SOM has greater application.

Future Directions

The SOM neural network is being proved superior to statistical techniques in pattern recognition, data mining, clustering, and its application in several areas including the ones mentioned in this article. Major areas, such as robotics, industrial and medical diagnostics, and geotechnology will see heavy application of SOMs in the future.

Cross References

► Patterns in Spatio-temporal Data

Recommended Reading

1. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall Inc, NJ (1999)
2. Pandya, A. S., Macy, R. B.: Pattern Recognition with Neural Networks in C++. CRC Press, Boca Raton, FL. (1996)
3. Lawrence, S., Giles, C.L., Tsoi, A.C., Back, A.D.: Face recognition: a convolutional neural network approach. IEEE Transactions on Neural Networks **8**(1), 1–27 (1997)
4. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. In: Intern. J. Comput. Vis.**1**(4), 321–331 (1987)
5. Hubel, D., Wiesel, T.: Receptive fields, binocular interaction & functional architecture in the cat's visual cortex. Journal of Physiology (London) **160**, 106–154 (1962)
6. Kohonen, T.: The self-organizing map. IEEE **78**, 1464–1480 (1990)
7. Kohonen, T.: Self-organizing Maps. Springer-Verlag, Inc., New York (1995)
8. Bhandarkar, S.M., Koh, J., Suk, M.: Multi scale image segmentation using a hierarchical self-organizing map. Neurocomputing **14**, 241–272 (1997)
9. Everly B: (2002) Personal communication. Product support, help@neuralware.com. Contacted 03.18.02

10. Kangas, J., Kohonen, T.: Development and applications of self-organizing map and related algorithms. Mathematics and Computers in Simulations **41**, 3–12 (1996)
11. Kohonen T: (2001) SOM-Tool Box Home. http://www.cis.hut.fi/projects/somtoolbox/. cited 02.04.01
12. Panda, S.: Data Mining Application in Production Management of Crop. Dissertation, North Dakota State University (2003)
13. Honkela T: (1997) Self-organizing map in natural language processing. http://citeseer.ist.psu.edu/123560.html. Accessed 30 Dec 2006

Self Organizing Map Usage

► Self Organizing Map (SOM) Usage in LULC Classification

Self-Referential Context

► Participatory Planning and GIS

Semantic

► Metadata and Interoperability, Geospatial

Semantic Discord

► Uncertainty, Semantic

Semantic Geospatial Web

► Geospatial Semantic Web, Interoperability

Semantic Information Integration

► Ontology-Based Geospatial Data Integration

Semantic Web

► Geospatial Semantic Integration
► Geospatial Semantic Web

Semivariogram Modeling

VIJAY GANDHI
Department of Computer Science,
University of Minnesota, Minneapolis, MN, USA

Synonyms

Variogram modeling

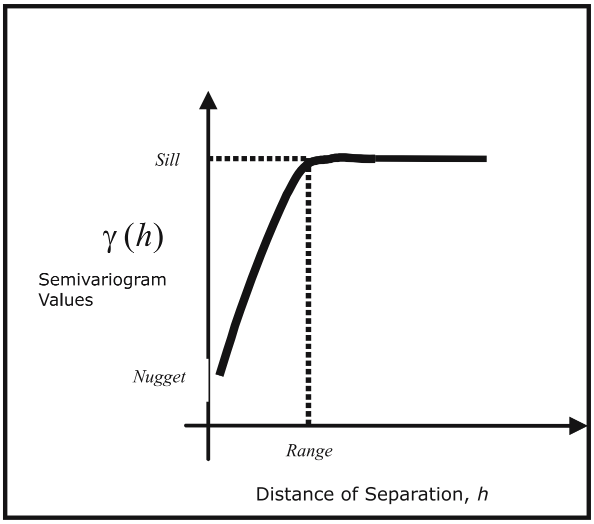
Definition

The first law of geography states that "Everything is related to everything else, but near things are more related than distant things". For example, it is natural that nearby places have more similar climates than do places which are far apart. Amounts of rainfall and iron ore deposits vary gradually over space. Such natural processes that vary gradually with respect to distance are said to be spatially correlated.

A semivariogram is one of the significant functions to indicate spatial correlation in observations measured at sample locations (Fig. 1). It is commonly represented as a graph that shows the difference in measure and the distance between all pairs of sampled locations. Such a graph is helpful in building a mathematical model that describes the variability of the measure with location. Modeling of the relationship among sample locations to indicate the variability of the measure with a distance of separation between all sampled locations is called semivariogram modeling.

In addition to summarizing the variation in measurements with distance, semivariogram modeling is also used as a prediction tool to estimate the value of a measure at a new location.

Semivariogram modeling is applied in fields related to spatial data, such as ecology (to study the vegetation cover), meteorology (to study the variation of climatic effects such as rainfall), and geology (to study the distribution of minerals such as iron ore and predict the iron ore content at a known location), etc.



Semivariogram Modeling, Figure 1 An example of a semivariogram

Semivariogram modeling is also referred to as variogram modeling.

Historical Background

The term "semivariogram" became popular after its first use by G. Matheron in 1963. Matheron used semivariogram modeling for the prediction of mining sites in South Africa with optimal ore grades. Such a prediction process, often referred to as kriging, is considered to be one of the first methods to consider spatial property. Because of such significant contributions, some call Matheron the father of spatial statistics.

Around the same time, Gandin developed ideas similar to the semivariogram. Work by Gandin and his colleagues was applied in the estimation of models for data related to the Soviet meteorological stations.

Semivariograms have been used in different fields and known by different names. In the study of probability, semivariograms have been referred to as a structure function by Yaglom, and as mean-squared difference in time series by Jowett.

Scientific Fundamentals

Consider x and $x+h$ to be locations where a measure, g, was taken. The locations $x+h$ and x are said be separated by h, which denotes the distance between the samples and the relative orientation. The mean difference between all such samples is:

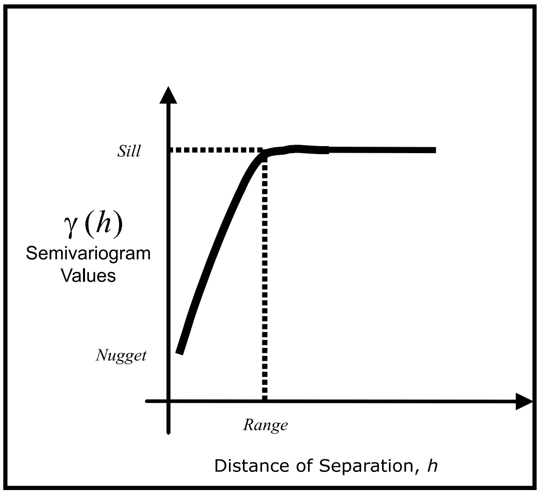
$$m(h) = \frac{1}{n} \sum [g(x) - g(x+h)] \,.$$

The variance corresponding to the above set of points is:

$$2\gamma(h) = \frac{1}{n} \sum [g(x) - g(x+h)]^2 \,.$$

The term $2\gamma(h)$ is called a variogram, and the term $\gamma(h)$ is called a semivariogram.

Semivariogram modeling is more of an exploratory data analysis technique. The values of $\gamma(h)$ are plotted against the difference in distances between pairs. Inferential results are obtained using the plotted graph. Figure 2 shows an "ideal model" of one such graph with a plot showing the change in $\gamma(h)$ with change in the distance of separation h. As can be inferred from Fig. 2, pairs of locations that are closer have smaller variance than pairs of locations that are farther apart. The variance gradually increases until a threshold is reached in the distance of separation. This threshold is called a *range*. Once the distance between two points is beyond a *range*, the variance becomes independent of the distance and maintains a constant value. Thus, the inverse of the *range* can be used to measure variability. The maximum variance value that can be attained by the variogram is called the *sill*. The *sill* of a model can be used to describe the variability as well. When the variogram is extrapolated to zero distance, the variance reaches a nonzero value called a *nugget*. Strictly speaking, this value should be zero when the distance between two points is zero. However, some factors such as sampling error may cause dissimilar values for samples at locations close to each other.



Semivariogram Modeling, Figure 2 Semivariogram, spherical model

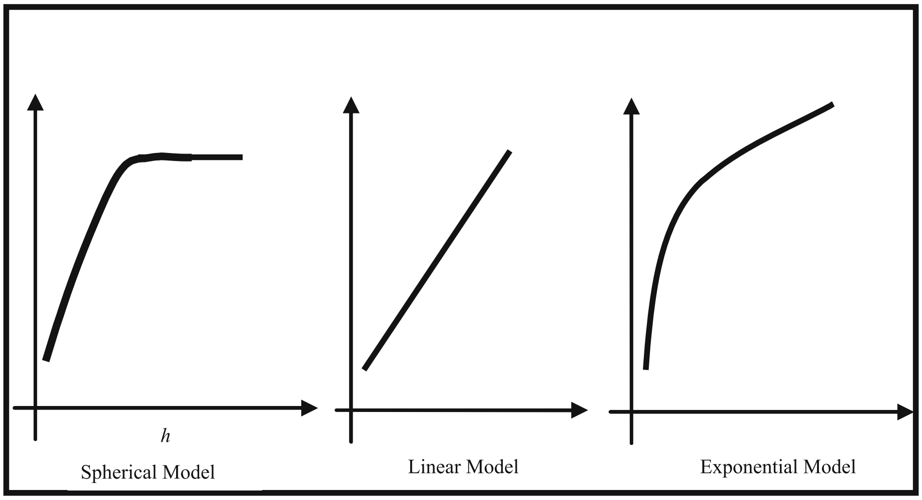
Semivariogram Models

The model described in the previous section is called a spherical model. Depending on the properties of the measure taken, a spherical model may not be the right fit for the dataset. Hence there is a need for other types of models. Some of the useful models include linear, spherical, exponential, and Gaussian models. Figure 3 shows the graphical representations of some of the semivariogram models. Table 1 lists those models and corresponding variogram functions. Some of the datasets may have complex behavior and may use a combination of such models.

Directional Semivariogram

The semivariogram discussed above considers only the distance of separation among all pairs of sample points. However, for data with spatial dependence, it might be logical to consider even the relative direction between pairs of sample points as well. Semivariograms for assessing anisotropy in data are called directional semivariograms. To consider direction, the region of sample points is divid-

S



Semivariogram Modeling, Figure 3 Variogram models

Semivariogram Modeling, Table 1 Variogram models

Variogram model	Variogram, γ (h)
Linear	$\gamma(h) = \begin{cases} \tau^2 + \sigma^2 h\,, & \text{if } h > 0 \\ 0\,, & \text{otherwise} \end{cases}$
Spherical	$\gamma(h) = \begin{cases} \tau^2 + \sigma^2\,, & \text{if } h >= \text{range} \\ \tau^2 + \sigma^2 \left[\frac{1.5*h}{\text{range}} - \frac{0.5*h^3}{\text{range}^3}\right], & \text{if } 0 < h <= \text{range} \\ 0\,, & \text{otherwise} \end{cases}$
Exponential	$\gamma(h) = \begin{cases} \tau^2 + \sigma^2(1 - \exp(-\phi h))\,, & \text{if } h > 0 \\ 0\,, & \text{otherwise} \end{cases}$
Gaussian	$\gamma(h) = \begin{cases} \tau^2 + \sigma^2(1 - \exp(-\phi^2 h^2))\,, & \text{if } h > 0 \\ 0\,, & \text{otherwise} \end{cases}$

ed into classes based on angles. Then, a semivariogram is plotted for each angle class.

Directional semivariogram are not so useful in assessing anisotropy because of the difficulty in interpreting results such as the variation of nuggets, sills and ranges with respect to the direction. Instead, other types of models such as rose diagrams and empirical semivariogram contour plots are used for assessing anisotropy.

Variogram Model Fitting

With a number of models available, it is important to select the one that is the most representative of the given dataset. Variogram model fitting procedures can be broadly divided into two categories: nonparametric models and parametric models.

Nonparametric Models

Ordinary Least Squares The traditional procedure to select the model involves plotting an empirical semivariogram i. e., the semivariogram using the sample data values. Then, a domain expert compares the empirical semivariogram with each of the theoretical models to find the closest match. Such a method is a crude way of finding the best variogram fit because for a given sample dataset, any number of theoretical models may seem to be proper. However, plotting of sample data is still useful as it provides a visual overview of the data. It may be used as an exploratory data tool to get some idea about the sample data.

Another way to select a model is to consider the problem as a curve-fitting problem and calculate the least-squares for each model. The model with the least error can be considered to be the best fit for the given data.

The empirical semivariogram is given by:

$$\gamma'(t) = \frac{1}{2N(t)} \sum_{(s_i,s_j)\in N(t)} [Y(s_i) - Y(s_j)]^2\,,$$

where, t indicates the distance of separation between a pair or points, $N(t)$ is the number of pairs with a maximum difference of t, s indicates any one of the spatial location, and $Y(s)$ indicates the value of a measure at location s.

The empirical semivariogram drawn using this equation mentioned above is considered to be a method of moments estimate, which is analogous to the sample variance estimate. This method has some drawback: it is sensitive to outliers and hence may have a bad average. Cressie and Hawkins [1] suggested a better estimate, using the following equation:

$$\gamma'(t) = \frac{1}{2N(t)} \sum_{(s_i,s_j)\in N(t)} |Y(s_i) - Y(s_j)|^{1/2} .$$

Once the empirical value, $\gamma'(t)$, is known, a least squares method is applied to test with the $\gamma''(t)$ value from the theoretical variogram. Now, according to the least squares method, the problem reduces to finding the variogram such that the sum-of-square error between $\gamma'(t)$ and $\gamma''(t)$ is minimum.

Parametric Models

Maximum Likelihood Maximum likelihood is a parametric model which assumes that the distribution model (e. g., Gaussian distribution is widely used in spatial statistics) of the data is known. The procedure then tries to retrieve the parameter values of the distribution model so as to fit the assumed distribution. The procedure, thus tries to select the values of the model parameters such that its likelihood is increased.

Restricted Maximum Likelihood The maximum likelihood procedure is considered to be expensive as it involves inversion of a matrix. Restricted maximum likelihood is an alternative method proposed by Cressie that reduces the computation.

Cross Validation

Cross validation is used to diagnose the variogram model selected. After fitting a variogram model, some of the data points are deleted and then the model selected is applied to predict the value at one of the deleted points. The lower the difference, the better is the selected model.

Key Applications

Semivariogram modeling is used in the prediction of unknown measure value at a known location. Such a method is called kriging [1]. By using information such as the nugget, sill, and range, the variogram function can be used to estimate (interpolate) the value at a known location. The values of nugget, sill and range determine the weights used in the process of kriging.

Variogram modeling is used in a number of fields associated with spatial data. Examples of such fields are ore mining, epidemiology, environmental modeling, image analysis, forecasting, etc.

Mineralogy

Prediction of ore content at a known site is probably the earliest application of variogram modeling [2]. It was first used by Matheron. In such an application, a variogram is modeled based on the ore quality at sample sites. The values of the variogram model are then used to estimate the value at a known location.

Meteorology

Variogram modeling can be useful statistical tool to study problems in atmospheric science. For example, it can be used to analyze the spatial distribution of rainfall at a known location by measuring the rainfall received only at sample sites.

Environmental Monitoring

It is natural to see the spatial correlation among many environmental factors such as soil type, land type, weather, etc. Hence, environmental monitoring is a common application which uses variogram modeling. Examples include modeling of spatial distribution of phosphor concentration in soil from samples, and distribution of scallops along a coast [3].

Epidemiology

Variogram modeling is used in epidemiology to study the spatial distribution of a particular disease [3].

Image Analysis

Variograms are used in satellite image classification. Texture is one of the features in image processing that represents the tonal variations in the spatial domain. Texture analysis provides characteristics related to variability patterns of landcover classes in digital image classification. A variogram is used to analyze the variability of texture [4].

Tool for Forecasting

Variogram modeling can also be used in time series for forecasting. It may be used to characterize the second-order dependence properties of a univariate time series, and in the construction of stationary time series models.

Future Directions

Semivariogram modeling is susceptible to errors and it may not be considered the best solution for a given dataset. It is more of an explanatory tool which is useful for getting a basic idea about the properties of the dataset. Advanced techniques based on Bayesian analysis are becoming more popular than semivariogram modeling.

Cross References

► Kriging
► Statistical Descriptions of Spatial Patterns

References

1. Cressie, N.: Statistics for Spatial Data. Wiley-Interscience, (1993)
2. Clark, I.: Practical Geostatistics. Applied Science, (1979)
3. Banerjee, S., Carlin, B.P., Gelfand, A.E.: Hierarchical Modeling and Analysis for Spatial Data, Chapmann & Hall/CRC, (2004)
4. de Jong, S.M., van der Meer, F.D.: Remote Sensing Image Analysis Including the Spatial Domain (Remote Sensing and Digital Image Processing). Springer, (2006)
5. Ma, C.: The use of the variogram in construction of stationary time series models. J. Appl. Probability 41, 1093–1103 (2004)
6. Chica-Olmo, M., Abarca-Hernandez, F.: Computing geostatistical image texture for remotely sensed data classification. Comput. Geosci. (1999)

Senses, Alternative

► Uncertainty, Semantic

Sensitivity Analysis

XIAOBO ZHOU, HENRY LIN
Department of Crop and Soil Sciences,
The Pennsylvania State University,
University Park, PA, USA

Synonyms

Sensitivity test; Susceptibility analysis; Sampling-based methods

Definition

Sensitivity analysis is the process of ascertaining how the change in the outputs of a given model depends upon the changes in model input factors (variables or parameters). A spatial sensitivity analysis is to apportion the variance of outputs of a spatial analysis or spatial prediction model to the variation of model input attributes, such as elevation, land use/land cover, erosion factor, and spatial data resolution. If a small change in input variables or model parameters results in a relatively large change in the output, the output is said to be sensitive to the variables or parameters. Sensitivity analysis is usually conducted via a series of tests in which the modeler uses different input values that vary around a central value within certain bounds to see how a change in input causes a change in the model output. By showing the dependence of a model on input variables or parameters, sensitivity analysis is an important tool in model design and model evaluation.

Historical Background

It is usually true that phenomena or processes in the real world are very difficult, or even impossible to quantitatively measure to absolute accuracy. Hence, parameters in dynamic spatial models characterizing such phenomena or processes have inherent uncertainty. Additionally, some parameters are not static properties, but dynamically change over time and/or space. Therefore, the parameter values of a spatial model are at least somewhat uncertain and eventually bring such uncertainty into the model output. Each model parameter usually contributes different uncertainties to the total uncertainty. It is good to know which parameters primarily affect model outputs and which parameters have little impact on model results during the model development process.

Sensitivity analysis provides a powerful tool to test the influence of the change in input factors (variables or parameters) on the change in the model output. In fact, the model development process should not be regarded as being complete without performing sensitivity analysis. If the tests reveal that the model is insensitive to an input factor, then it may be possible to use an estimate rather than a value with greater precision for that factor. This is particularly useful when acquiring a precise value is costly and/or time-consuming. It also gives more confidence to the modeler about the uncertainty of the model behavior for that factor. If the tests indicate that the model behavior greatly depends on the change of factor, then the sensitive parameter must be carefully determined to make the model more realistic and valid.

By varying the input factors within a wide range, a sensitivity test gives modelers an opportunity for a better understanding of the dynamic behavior of the system, particularly the behavior of a system in extreme situations. Sensitivity analysis allows modelers to determine the level of accuracy that is necessary for an input factor to make the model sufficiently useful and valid. A comprehensive sensitivity analysis not only identifies the sensitive/insensitive factors, but also finds the sensitive/insensitive ranges for a sensitive factor. Reasonable values to be used for input factors in the model can be suggested via a sensitivity analysis.

Sensitivity analysis was originally created to investigate the influence of input factors on output uncertainty. Recently, the applications of sensitivity analysis have been extended to test model quality. A well-developed model should not a show strong response to the assumed insensitive factors, but will respond to sensitive ones. Otherwise, the model needs to be reexamined and eventually revised, including its assumptions, structures, specifications, and others.

Scientific Fundamentals

There are a number of methods for performing sensitivity analysis. The choice of a sensitivity analysis method depends to a great extent on three aspects: the sensitivity indices employed, the desired accuracy of the sensitivity indices, and the computational cost involved. Generally, sensitivity analysis methods can be grouped into three classes: screening design method, *local sensitivity analysis*, and *global sensitivity analysis* (see Cross References). The most common and simplest sensitivity analysis is a sampling-based method that involves running the model repeatedly for a combination of factor values sampled from the distribution of the input factors, and establishing the relationship between inputs and outputs. Sampling-based methods do not require access to model equations. The sampling-based method involves five basic steps, as follows [1]:

- Design experiment and select the input of interest
- Assign distribution functions or ranges of variation to selected factors
- Generate a matrix of inputs
- Evaluate the model and compute the distribution of the target function
- Select a method to assess the influence of each input factor on the output

The sensitivity analysis can also be performed based on groups instead of individual input factors by dividing the input factors into groups according to knowledge of the factors. As a result, the computation cost of the analysis is greatly reduced.

Key Applications

Sensitivity analysis provides information to modelers and decision makers about the main factors that contribute to the overall uncertainty in model outputs, including model structure, assumptions, and parameter specifications. The power of sensitivity analysis has been demonstrated in a wide variety of application areas [1,2].

Spatial Analysis

Sensitivity analysis is an important technique for spatial analysis including spatial interpolation, density estimation, spatial autocorrelation and spatial association. A detailed spatial sensitivity analysis provides useful information on the impacted area and spatial trend as the result of changing input factors. For example, spatial sensitivity analysis of buffer zone area around sample locations can be used to determine the relationships between land use and nutrient or pollutant concentrations and the impaired range. Sensitivity analysis can be also applied to investigate the sensitivity of spatial autocorrelation to different spatial weight matrixes, and to assess the detected cluster and spatial outliers.

Geosciences and Hydrology

Sensitivity analysis is a key component in developing models of water flow and contaminant transport through porous media including the Earth's crust and mantle [3]. For example, it allows one to forecast water content, solute concentration, temperature, and velocity distributions in the mantle even without a perfect knowledge of the initial conditions. The relative sensitivity of leachate transport control parameters, e. g., source strength, hydraulic conductivity, and dispersivity can be determined and incorporated in simulating leachate generation and transportation via a sensitivity analysis.

Experiment Design and Data Collection

Sensitivity analysis is very helpful for experiment design by providing an a priori evaluation of the stability and establishing the allowable level of uncertainty. Sensitivity analysis is beneficial in determining the direction of data collection activities. Data to which the model is relatively sensitive would require more characterization, as opposed to data to which the model is relatively insensitive. Model-insensitive data may not require further field characterization.

Agriculture

In agriculture, sensitivity analysis can be used to identify the most efficient way to increase crop yield, for example, tillage practice, fertilizer application, and pest control. A sensitivity analysis can also measure the impact that changes in any factor (yields, prices, sales, costs, interest rate, etc.) will have on net farm income, and identify the factor that affects profitability the most. Such sensitivity analysis increases farmer's confidence in profitability projections.

Environmental Science

A *global sensitivity analysis* can be used in decision making to balance the influence of various uncertain factors. By investigating the contributions of input variables to the overall uncertainty in the risk assessment, sensitivity analysis can help set priorities among sites, contaminants, or exposure pathways. The input parameters with the highest priority for further investigation are those that cause the greatest effect on the overall uncertainty of the risk. Sensitivity analysis can highlight key factors for further examination and indicate the potential management strategy and remediation practice that will lead to the quickest recovery for polluted areas, and hence is an indispensable tool for the management of uncertainties in models for environmental regulations and policy.

Biology

Sensitivity analysis provides insights into the nature and function of biochemical systems, improves our understanding of the population dynamics, and serves as a tool to predict the result of interventions. A sensitivity analysis can assess the relative importance of various factors and physical processes involved in complicated biological and ecological systems. Sensitivity analysis of oscillatory systems (e. g., period and amplitude) is also an area of active research [4].

Control System

The oldest applications of sensitivity analysis are in control theory [5]. Sensitivity analysis is a crucial aspect of control systems, particularly useful for the investigation of control system stability and physical realizability. Sensitivity analysis techniques can be used to deal with parameter specifications for the worst-case conditions in the form of tolerances, and permit the minimization of the control system sensitivity with respect to the parameters.

Business

Sensitivity analysis can provide information to managers about the change of which elements of the business primarily affects the company's profit performance and hence requires more attention [4]. For example, a scenario might be to determine which business element will bring the most effect on profit if sales, cost of goods sold, operation expenses, or interest rates were to increase or decrease by 10%. By evaluating a variety of "what if" questions, managers can look at the potential effect of undesirable outcomes such as a shortfall in sales or an overrun on costs.

Future Directions

Sensitivity analysis is a must-do step for model building or model evaluation. It is often the case that steady-state behavior is of primary interest. As a result, sensitivity analysis is typically applied for the system at or near the steady state. In order to understand the mechanisms behind dynamic behaviors, however, there is a need to extend the classical (steady-state) sensitivity analysis.

Data collection is the most important and expensive part of geographic information systems (GIS). Since sensitivity analysis provides information on the sensitivity of input factors on the model results, it may also act as a resource optimization process. The development of enhanced GIS toolboxes to assist in performing and displaying the results of various types of sensitivity analysis will also be beneficial in future spatial analysis.

Cross References

► Global Sensitivity Analysis
► Local Sensitivity Analysis
► Screening Method

Recommended Reading

1. Saltelli, A., Chen, K., Scott, E.M.: Sensitivity Analysis. Chichester, New York (2000)
2. Cacuci, D.G.: Sensitivity and Uncertainty Analysis, Vol. 1: Theory. Chapman and Hall, Boca Raton, FL (2003)
3. Hier-Majumder, C.A., Travis, B.J., Belanger, E., Richard, G., Vincent, A.P., Yuen, D.: A.: Efficient sensitivity analysis for flow and transport in the Earth's crust and mantle. Geophys. J. Int. **166**, 907–922 (2006)
4. Ihekwaba, A.E.C., Broomhead, D.S., Grimley, R.L., Benson, N., Kell, D.B.: Sensitivity analysis of parameters controlling oscillatory signaling in the NF-kB pathway: the roles of IKK and IkBa. Syst. Biol. **1**, 93–103 (2004)
5. Santner, T. J., Williams, B. J., Notz, W.I.: The Design and Analysis Of Computer Experiments. Springer-Verlag, New York(2003)
6. Rios-Insua, D.: Sensitivity Analysis In Multi-Objective Decision Making. Springer-Verlag, Berlin (1990)
7. Saltelli, A., Tarantola, S., Campolongo, F., Ratto, M.: Sensitivity Analysis in Practice: a Guide to Assessing Scientific Models. Wiley, Hoboken, NJ (2004)

Sensitivity Test

► Sensitivity Analysis

Sensor Networks

► Geosensor Networks
► Geosensor Networks, Estimating Continuous Phenonena

Sensor Orientation

▶ Photogrammetric Methods

Separability

▶ Hurricane Wind Fields, Multivariate Modeling

Sequential Patterns, Spatio-temporal

YAN HUANG, LIQIN ZHANG
Department of Computer Science and Engineering,
University of North Texas, Denton, TX, USA

Synonyms

Mining sequential patterns from spatio-temporal databases

Definition

Space and time are pervasive in everyday life and technology is changing the way they are tracked. With the advances of technologies such as GPS, remote sensing, RFID, indoor locating devices, and sensor networks, it is possible to track spatio-temporal phenomena with increasingly finer spatial resolution for longer periods of time. Depending on the characteristics of available spatio-temporal datasets, sequential patterns are defined in two ways. When trajectory datasets for moving objects are given, the sequential pattern mining problem can be defined as: Given the spatio-temporal trajectory of a moving object $\{\{x_1, y_1, t_1\}, \{x_2, y_2, t_2\}, \ldots, \{x_n, y_n, t_n\}\}$ and a support *min_sup* [2,3,11], find frequent sub-trajectories in the form of $r_1 r_2 \ldots r_q$, that appears more then *min_sup* times where r_k is a region after clustering point trajectory data into regions. These patterns are called **Trajectory Based Sequential Pattern (TBSP)**. Such patterns are routes frequently followed by an object [3]. An example TBSP pattern may be a moving pattern such as Home → Sack and Save → Walmart → Best Buy → Home as a frequent shopping routine in Denton, Texas.

When spatio-temporal events instead of trajectories are given, the problem becomes a **Event Type Based Sequential Pattern (ETBSP)** problem. An event happens at a given place and time and belongs to a specific event type. An event type categorizes or groups events that have the same characteristics together. Example event types include car accidents, traffic jams, and the West Nile disease. For this problem, it is necessary to find the sequential pattern among *different event types* [9], instead of locations/regions. For example, it is possible to observe the West Nile [6] transmission path as "Bird → Mosquito in nearby region in a day → Human being in nearby region in two days". Given a database $\mathcal{D}$ of spatio-temporal events, let $\mathcal{F} = \{f_1, f_2, \ldots, f_K\}$ be a set of event types. Each event has the following fields: *event-id, time, location*, and *event-type*, where *event-type* $\in \mathcal{F}$. The problem of finding spatio-temporal sequential patterns is to find all the *significant* event type sequences in the form of $f_1 \to f_2 \ldots \to f_k$.

Historical Background

Mining sequential patterns from market basket data has attracted much attention since introduced by Agrawal et al. [1]. It addresses the following problem. Given customers' purchasing records, the objective is to find common purchased subsequences shared by a significant number of customers, e. g., (PC, Internet service) → DVD → MP3 player. Many efficient algorithms [12,14] have been proposed to identify sequential patterns from the market basket data. However, the "transactionization" of spatio-temporal space to adapt the existing sequential pattern mining methods on market basket data is a non-trivial task and may be unnatural due to the continuity of spatial and temporal space [8].

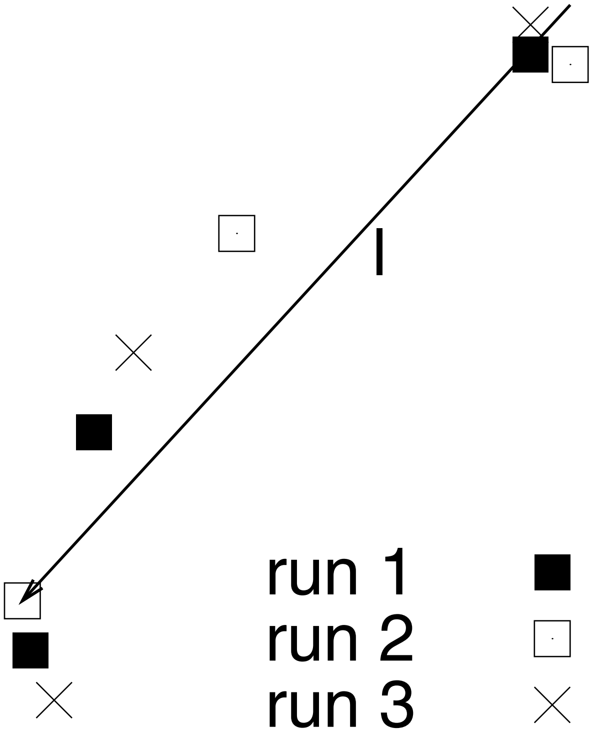
The trajectory of a moving object is typically a collection of spatial signatures at consecutive time stamps. Retrieving similar trajectories might reveal underlying traveling patterns of moving objects in the data. Mamoulis et al. discussed mining, indexing, and querying of historical spatio-temporal data in [2,11]. In their context, trajectories of objects are given for investigations. Models and algorithms for mining frequent *periodic* sub-trajectories were proposed. However, the patterns found depend on the fixed period T and, additionally, the instances of pattern may be distorted or shifted, i. e., events may shift greatly so that it can't fall in the same regions. Cao et al. presented an algorithm for finding shifted frequently repeated sequences [3]. Trajectory is the collection stops of the **same** moving object at different spatial locations. Trajectory analysis can be applied only if the trajectories have been provided a priori. When the spatio-temporal dataset is a collection of different events, with each event belonging to one particular event type, trajectory may not be available for mining spatio-temporal sequential patterns in this context. The event type based sequential pattern mining problem was modeled in [9] to deal with mining sequential patterns from spatio-temporal events.

S

Scientific Fundamentals

Trajectory Based Sequential Patterns

The recent developments [3] of TBSP based sequential patterns are summarized in this section. In TBSP, locations



Sequential Patterns, Spatio-temporal, Figure 1 Object Movement

are not repeated *exactly* in every instance of a movement pattern. The main idea in [3] is to first summarize a series of spatial locations to spatial regions. Motivated by line simplification techniques[5], the segments of spatio-temporal location series were represented by directed line segments. Figure 1 shows that the line segment l summarizes the first three points in each of the three runs with small errors. In this way, not only can the original data be compressed, decreasing the mining effort, but also the derived line segments (which approximately describe movement) provide initial seeds for defining the spatial regions which could be expanded later by merging similar and nearby segments.

A **segment** s_{ij} in a spatio-temporal sequence S $(1 \leq i < j \leq n)$ is a contiguous subsequence of S, starting from (x_i, y_i, t_i) and ending at (x_j, y_j, t_j). Given s_{ij}, we define its **representative line segment** $\vec{l_{ij}}$ with starting point (x_i, y_i) and ending point (x_j, y_j). Let ϵ be a distance error theshold, s_{ij} **complies with** $\vec{l_{ij}}$ with respect to ϵ and is denoted as $s_{ij} \propto_\epsilon \vec{l_{ij}}$ if $\text{dist}((x_k, y_k), \vec{l_{ij}}) \leq \epsilon$ for all $k (i \leq k \leq j)$, where $\text{dist}((x_k, y_k), \vec{l})$ is the distance between x_k, y_k and line segment $\vec{l}$. When $s_{ij} \propto_\epsilon \vec{l_{ij}}$, each point (x_k, y_k), $i \leq k \leq j$, in s_{ij} can be projected to a point $(x_{k'}, y_{k'}$ on $\vec{l_{ij}}$. $x_{k'}, y_{k'}$ implicitly denotes the projection of x_k, y_k to $\vec{l_{ij}}$.

A **segmental decomposition** S^s of S is defined by a list of consecutive segments that constitute S. Formally, $S^s = s_{k_0k_1} s_{k_1k_2} \ldots s_{k_{m-1}k_m}$, $k_0 = 1, k_m = n, m < n$, where $s_{k_ik_{i+1}} \propto_\epsilon \vec{l_{k_ik_{i+1}}}$ for all i, to simplify notation, $s_0 s_1 \ldots s_{m-1}$ is used to denote S^s.

Let $\vec{l}$ represent a directed line segment, and let $\vec{l}.angle$ and $\vec{l}.len$ be the slope angle and length, respectively. Two line segments $\vec{l_{ij}}$ and $\vec{l_{gh}}$ representing segments s_{ij} and s_{gh} are **similar**, denoted by $\vec{l_{ij}} \sim \vec{l_{gh}}$, with respect to the angle difference threshold θ and length factor f $(0 \leq f \leq 1)$ if:

(i) $|\vec{l_{ij}}.angle - \vec{l_{gh}}.angle| \leq \theta$ and

(ii) $|\vec{l_{ij}}.len - \vec{l_{gh}}.len| \leq f \times max(\vec{l_{ij}}.len, \vec{l_{gh}}.len)$ if $\vec{l_{ij}} \sim \vec{l_{gh}}$, s_{ij} and s_{gh} are also treated as similar to each other.

Note that similarity is symmetric.

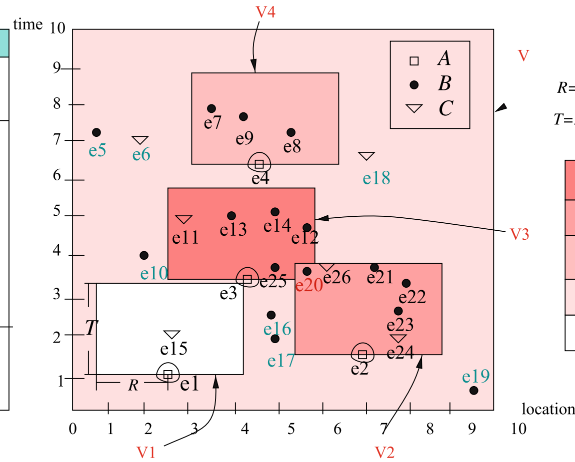
Line segment $\vec{l_{ij}}$ is **close** to $\vec{l_{gh}}$ if for $\forall (x'_k, y'_k) \in \vec{l_{ij}}$, $dist((x'_k, y'_k), \vec{l_{gh}}) \leq \epsilon$ (ϵ is the distance error threshold). When $\vec{l_{ij}}$ is close to $\vec{l_{gh}}$, it also means that the segment s_{ij} is close to the segment s_{gh}, where $s_{ij} \propto_\epsilon \vec{l_{ij}}$ and $s_{gh} \propto_\epsilon \vec{l_{gh}}$. As opposed to similarity, closeness is asymmetric.

Let L be a set of segments from sequence S^s. The **mean line segment** for L, $\vec{l^c}$, is a line segment that best fits all the points in L with the minimum sum of squared errors (SSE). Let tol be the average orthogonal distance of all the points in L to $\vec{l^c}$. A spatial pattern element is a rectangular spatial region r_L with four sides determined by $(\vec{l^c}, tol)$ as following: (1) two sides of r's that are parallel to $\vec{l^c}$ have the same length as $\vec{l^c}$ and their distances to $\vec{l^c}$ are tol; (2) the other two vertical sides have length $2 \cdot tol$, and their midpoints are the two end points of $\vec{l^c}$. $\vec{l^c}$ is referred as the **central line segment** of region r_L. Region r_L contains k segments or k segments contribute to r_L if L consists of k segments. A **spatio-temporal sequential pattern** P is an ordered sequence of pattern elements: $r_1 r_2 \ldots r_q$, $(1 \leq q \leq m)$. The **length** of pattern P is the number of regions in it.

A contiguous subsequence of S^s, $s_i s_{i+1} \ldots s_{i+q-1}$, is a **pattern instance** for P: $r_1 r_2 \ldots r_q$ if $\forall j (1 \leq j \leq q)$, if the representative line segment for segment s_{i+j-1} is *similar* and *close* to the central line segment of region r_j. P' is a **subpattern** of P if $i \leq j$ and $\exists k$, $(1 \leq k \leq j - i + 1)$ such that $r'_1 = r_k, r'_2 = r_{k+1}, \ldots, r'_i = r_{k+i-1}$. P is a **superpattern** of P'.

The **support** of a pattern P is the number of instances supporting P. Given a support threshold *min_sup*, P is **frequent** if its support exceeds *min_sup*. Since a pattern with the same frequency to one of its supersets is redundant, the mining problem focuses on detecting *closed* frequent patterns [7], for which every proper subpattern has equal frequency. The mining **problem** is to find frequent patterns from a long spatio-temporal sequence S with respect to a support threshold *min_sup*, and subject to a segment-

Event Type	Symbol	Event Set (Location, Time)
A	□	e1(2.55, 1.05), e2(6.90, 1.55), e3(4.20, 3.60), e4(4.50, 6.40)
B	•	e5(0.70, 7.20), e7(3.50, 7.90), e8(5.20, 7.20), e9(4.20, 7.70), e10(2.00, 4.00), e12(5.60, 4.70), e13(3.90, 5.00), e14(4.90, 5.10), e16(4.80, 2.60), e17(4.90, 1.90), e19(9.55, 0.60), e20(5.60, 3.70), e21(7.20, 3.80), e22(7.90, 3.45), e23(7.80, 2.80), e25(4.90, 3.80)
C	▽	e6(1.95, 7.00), e11(2.90, 4.90), e15(2.60, 2.00), e18(7.00, 6.60), e24(7.80, 1.95), e26(6.10, 3.80)



Sequential Patterns, Spatio-temporal, Figure 2 *Left*: A sample spatio-temporal dataset; *right*: densities of events of *B* in events of *A*s' neighborhoods and overall are represented by shades of different intensities

ing distance error threshold ϵ, a similarity parameter θ and a length factor f.

A frequent trajectory mining algorithm has three phases: In the first step, it simplifies the point data into line segments, then, in the second step, it clusters the line segments into regions which are used as the frequent singular patterns in the third step. Hence, the mining problem is transferred into sequential patterns among regions.

Event Type Sequential Patterns

This section introduces basic concepts, significant measures, and mining algorithms of ETBSP. For the simplicity of illustration, the dimensionality of both space and time is assumed to be one. A spatio-temporal predicate called *"follow"* is defined and used to explain the significance measure and how to mine the event type based sequential patterns. However, the framework is general and can accommodate other spatio-temporal predicates as long as the predicate can be described by a well defined neighborhood definition. Informally, the *"follow"* spatio-temporal predicate describes that an event e_1 happens in the nearby region of another event e_2 shortly after e_2 happened.

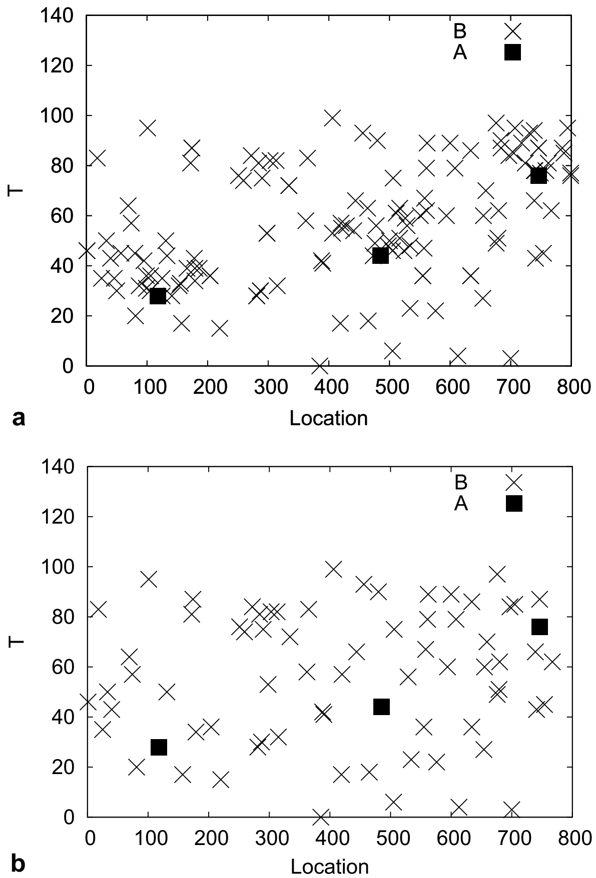
Follow: *An event e' follows$_N$ a different event e, denoted by $e \rightarrow_N e'$, if and only if e' is in the spatio-temporal neighborhood $N(e)$ of e. A simple spatio-temporal neighborhood of an event e to capture the "follow" predicate can be defined as $N(e) = \{p|distance(e.location, p.location) \leq R \wedge (p.time - e.time) \in [0, \mathcal{T}]\}$.*

An example of the simple neighborhood could be a distance less than 1.5 miles for any time interval less than 1 hour. When the neighborhood definition for the "follow" predicate is clear and causes no confusion, $e \rightarrow_N e'$ is abbreviated to $e \rightarrow e'$.

In Fig. 2, three spatio-temporal event types are represented using the symbols square (A), circle (B) and triangle (C), respectively. Their events are embedded in a spatio-temporal space. The x-axis denotes the linear space and the y-axis denotes the linear time. The event set for each event type is listed in a table on the left side of the figure. Among many others, the event e_{11} follows the event e_3, i. e., $e_3 \rightarrow e_{11}$ for the given distance $\mathcal{R}$ and time interval $\mathcal{T}$ as shown in the figure. Note that the definition of the spatial distance $\mathcal{R}$ can be applied to both left and right directions in space while the definition of the temporal interval $\mathcal{T}$ can only be applied to upward direction due to the one directional property of time (this property can be utilized for designing efficient algorithms [9]).

One way to define the significance measure of a pattern $f_1 \rightarrow f_2$ is to calculate **the number** of events of type f_2 that follow events of type f_1. However, this approach ignores the distribution of events of f_2 in the overall space. For example, in Fig. 3b, when event type B has many events, this naive significance measure will mistakenly conclude that since there are many events of B following events of A, then $A \rightarrow B$. The main problem of simple counting based measures is that they do not compare with randomness and provide little guidance on how to determine a proper threshold for the counts to define significance. As a result, spurious sequences may be detected as significant sequential patterns especially when the numbers of events of involved event types are large. Instead of using a simple counting mechanism, **sequence index** is a significance measure for spatio-temporal sequential patterns which may be more meaningful due to its interpretability using spatial statistics [4].

The concept of *density* is defined first to facilitate the definition of the density ratio. The density of an event set in



Sequential Patterns, Spatio-temporal, Figure 3 **a** B follows A. **b** B is independent of A

a selected spatial-temporal space V is the average number of events in a unit of V, i. e., the total number of events of the event set divided by the volume $|V|$ of selected spatio-temporal space V. The selected spatio-temporal space may be the whole embedding space or some arbitrary neighborhood, e. g., the neighborhood of an event.

In Fig. 2, the density of the event set of the event type B in the whole spatio-temporal embedding space V is $\frac{16}{|V|} = 0.16$ since $|V| = 10 \times 10 = 100$, meaning in each unit of the spatio-temporal embedding space, are expected to be found 0.16 events of type B. The density for the event set of the event type B in region V_4 is $\frac{3}{|V_4|} = 0.3$, where $|V_4|$ is the volume of neighborhood of e_4. Given $\mathcal{R} = 2$ and $\mathcal{T} = 2.5$, $|V_4| = 2 \times 2 \times 2.5 = 10$. Using the density concept, the density of an event type f in space V may be represented by $density(f.\mathcal{E}, V)$, where $f.\mathcal{E}$ is the event set of f.

Given two event sets E and E', if the average density of E' in the neighborhoods of events in E is higher than the density of E' in the overall embedding space, it is likely that events in E' tend to follow events in E. Density Ratio is used to measure the percentage of E' following E. Formally, the density ratio concept is defined as:

Density Ratio: *For two event sets E and E′, and a given neighborhood function $N(e)$ around any event e, the density ratio of $E \rightarrow E'$ is defined as: $densityRatio(E \rightarrow E') = \frac{average_{e \in E}(density(E', N(e)))}{density(E', V)}$, where V is the spatio-temporal embedding space.*

A spatio-temporal sequence of k event types is called a (k)-sequence. The subsequence consisting of ith to jth event types of a (k)-sequence $\vec{S}$ is referred to as $\vec{S}[i : j]$ with $i \in [1, k-1], j \in [i+1, k]$ and the ith event type in a sequence $\vec{S}$ as $\vec{S}[i]$. Generalizing density ratios to longer-sequences $(k > 2)$ is non-trivial. For a sequence $\vec{S}$ to be significant, there are two requirements that the significance measure needs to meet:

1. The events of any event type $\vec{S}[i]$ in $\vec{S}$ need to "*follow*" an "event sequence" of the sequence $\vec{S}[1 : i-1]$.
2. The density ratio between any two consecutive event types needs to be significant.

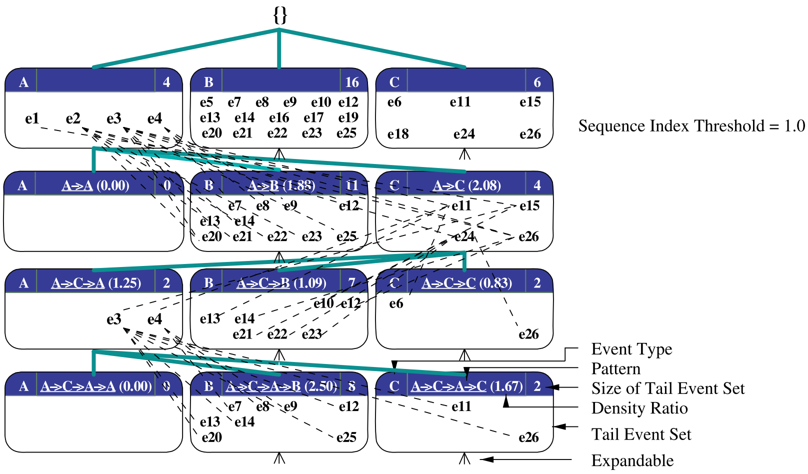
To meet the first requirement, the concept of event sequence and a tail event set of a (k)-sequence are defined. A tail event set serves as a link to capture the *follow* relationship between an event type with a sequence.

Event Sequence: *An event sequence $\vec{E}$ of a sequential pattern $\vec{S}$ is a sequence of events such that: (1) $\vec{E}[i]$ is of event type $\vec{S}[i]$, for $i \in [1, k]$; (2) $\vec{E}[i] \rightarrow \vec{E}[i+1]$, for $i \in [1, k-1]$. An event e of event type $\vec{S}[k]$ is said to "follow" the sequence $\vec{S}[1 : k-1]$, if there exists an event sequence $\vec{E}'[1 : k-1]$ of $\vec{S}[1 : k-1]$ such that appending e to $\vec{E}'$ produces an event sequence of $\vec{S}[1 : k]$.*

Tail Event Set of a Sequence: *For a (k)-sequence $\vec{S}$, the set of events of type $\vec{S}[k]$ that participates in any event sequence of S is called the tail event set of $\vec{S}$ and is denoted as tail_event_set($\vec{S}$).*

To meet the second requirement of the significance measure for a sequence $\vec{S}$, a sequence index is defined to be the minimal of the density ratio between the tail event set of an $(i-1)$-subsequence $\vec{S}[1 : i]$ and the event set of event type $\vec{S}[i]$, i. e., $densityRatio(tail\_event\_set(\vec{S}[1 : i-1]) \rightarrow \vec{S}[i].\mathcal{E}))$, where $i \in [1, k-1]$. Instead of using all of the events in $\vec{S}[i-1].\mathcal{E}$, the tail event set of $\vec{S}[1 : i-1]$ is used to compute the density ratio with $\vec{S}[i].\mathcal{E}$ because the goal is to measure the impact of the $(k-1)$-sequence on the event type $\vec{S}[i]$, instead of the impactof the event type $\vec{S}[i-1]$. Now, density ratio is used to define sequence index:

Sequence Index: *The sequence index of a (k)-sequence $\vec{S}$ is defined as:*



Sequential Patterns, Spatio-temporal, Figure 4 Partial Sequential Pattern Tree for the Sample Data in Fig. 2

1. *When* $k = 2$, $seqIndex(\vec{S}) = densityRatio(\vec{S}[1].\mathcal{E} \rightarrow \vec{S}[2].\mathcal{E})$;
2. *When* $k \geq 3$, $seqIndex(\vec{S}) = \min(seqIndex(\vec{S}[1:k-1], densityRatio(tail\_event\_set(\vec{S}[1:k-1]) \rightarrow \vec{S}[k].\mathcal{E}))$

A (k)-sequence $\vec{S}$ is called a *significant* spatio-temporal sequence if $seqIndex(\vec{S}) \geq \theta$, where θ is a user given minimum sequence index. *The problem of mining spatio-temporal sequential patterns is to find all significant spatio-temporal sequences.*

To mine the sequential patterns efficiently, a Sequential Pattern Tree is built. An example of the tree is shown in Fig. 4. The tree is initially a one node, and then the tree is expanded in a depth first manner. Any time only a subtree rooted at E is in memory. The tree is updated when an event sequence can be expanded. Without loss of generality, let us assume $b \in B$, and $a \in P$, where P is a pattern ended with A. After finding all the event sequences of the potential pattern $P \rightarrow B$, the significance of the pattern can be verified by computing its sequence index. If the computed value is larger than the given threshold, $P \rightarrow B$ is a new pattern that expand P with one more event type B. Now, this pattern can be further expanded in a similar manner. The algorithm stops when the sequence index drops below the given threshold. If the dataset is large, the data can be sliced into pieces and only one slice is brought into memory each time. For each slice, it is processed using the same method above, however, the sequence crossing boundaries of slices will need to be "stitched" together and the whole tree needs to be in memory and the given threshold is set to 0. After processing all slices, the patterns can be verified from the tree by computing the densities indexes.

Key Applications

Trajectory Sequential Patterns

Parts of the object routes are often repeated in the archived history of locations. For instance, buses move along series of streets repeatedly, people go to and return from work following more or less the same routes, etc. However, the movement routes of most objects (e. g., private cars) are not predefined. Even for objects (e. g., buses) with prescheduled paths, the routes may not be repeated with the same frequency due to different schedules on weekends or on special days. Finding frequently repeated paths, i. e., *spatio-temporal sequential patterns*, from a long spatio-temporal sequence is very helpful for the following applications:

Prediction: These patterns could help to analyze/predict the past/future movement of the object and support approximate queries on the original data.

Schedule Arrangement: These patterns could also support schedule arrangement. For example, the traveling patterns of visitors can be found. Such patterns can allow agents to arrange their schedules in order to attract more passengers, e. g., an airline company can schedule a flight as long as there are enough passengers and arrange the departure time for the next location in the sequential pattern properly after the arriving time of the previous one.

Event Type Sequential Patterns

Many spatio-temporal events interact with each other and tend to exhibit spatial and temporal patterns which may be summarized in event type level. It is critical to characterize such interactions involving space and time together among event types and identify them from large spatio-temporal datasets efficiently in various application domains. These domains include Earth science, epidemiology, ecology, and climatology [10,13,15]. Two of the applications are discussed as follows:

Earth Science: With earth science data available, it is possible to mine the sequential pattern from different features such temperature, Net product, drought index, and UV index. With the knowledge of such patterns, for example, high temperature → high net products for some special plant, further investigations can be made about the possible causalities of the events of interest.

Epidemiology: ETBSP sequential patterns may help us to gain useful information to understand the transmission of disease. In the West Nile example, if "Bird → Mosquito → Human being" is found as a valid sequential pattern, it may be necessary to cut the transmission of the disease from bird to human by controlling mosquitos.

Future Directions

With the increasing popularity of location tracking devices such as GPS and geo-sensor networks, trajectory and spatio-temporal event datasets will continue to grow in the near future. New interesting patterns such as "flock" representing a group of entities that "travel" together for an extended period of time will continue to emerge. Defining more meaningful patterns that are useful for a broad range of applications will be a future direction. The other direction may be the database support for mining these patterns. New language constructs that are succinct, less intrusive to current database languages, and semantically meaningful for a broad range of applications using spatio-temporal sequential patterns need to be developed. Query processing and query optimization techniques need to be investigated as well.

Recommended Reading

1. Agrawaland, R., Srikant, R.: Mining sequential patterns. In: Proc. of SIGKDD (1995)
2. Cao, H., Cheung, D.W., Mamoulis, N.: Discovering partial periodic patterns in discrete data sequences. In: Proc. of PAKDD (2004)
3. Cao, H., Mamoulis, N., Heung, C.D.W.K.: Mining frequent spatio-temporal sequential patterns. In ICDM (2005)
4. Cressie, N.A.C.: Statistics for Spatial Data. Wiley and Sons, ISBN:0471843369 (1991)
5. Douglas, D.H., Peucker, T.K.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. In: The Canadian Cartographer, vol. 10, No.2, pp. 112–122 (1973)
6. Centers for Disease Control and Prevention (CDC). http://www.cdc.gov/ncidod/dvbid/westnile. 11 Sept 2007
7. Han, J., Wang, J., Lu, Y., Tzvetkov, P.: Mining top-k frequent closed patterns without minimum support. In: Proc. of Intl. Conf. on Data Mining, pp. 211–218. 9–12 Dec 2002
8. Huang, Y., Shekhar, S., Xiong, H.: Discovering Co-location Patterns from Spatial Datasets: A General Approach. IEEE TKDE, **16**(12) 1472–1485 (2004)
9. Huang, Y., Zhang, L., Zhang, P.: Finding sequential patterns from a massive number of spatio-temporal events. In: Proc. of SIAM International Conference on Data Mining (SDM) (2006)
10. Koubarakis, M., Sellis, T.K., Frank, A.U., Grumbach, S., Güting, R.H., Jensen, C.S., Lorentzos, N.A., Manolopoulos, Y., Nardelli, E., Pernici, B., Schek, H.-J., Scholl, M., Theodoulidis, B., Tryfona, N.: Spatio-Temporal Databases: The CHOROCHRONOS Approach. Springer, New York (2003)
11. Mamoulis, N., Cao, H., Kollios, G., Hadjieleftheriou, M., Tao, Y., Cheung, D.W.L.: Mining, indexing, and querying historical spatiotemporal data. In: Proc. of SIGKDD (2004)
12. Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Wang, J., Chen, Q., Dayal, U., Hsu, M.-C.: Mining sequential patterns by pattern-growth: The prefixspan approach. Proc. of SIGKDD (2004)
13. Roddick, J.F., Spiliopoulou, M.: A Bibliography of Temporal, Spatial and Spatio-temporal Data Mining Research. ACM Special Interest Group on Knowledge Discovery in Data Mining (SIGKDD) Explorations, http://kdm.first.flinders.edu.au/IDM/STDMBib.html (1999). Accessed 11 Sept 2007
14. Zaki, M.: SPADE: An efficient algorithm for mining frequent sequences. Machine Learning, **42**(1/2):31–60 (2001)
15. Zhang, P., Steinbach, M., Kumar, V., Shekhar, S., Tan, P., Klooster, S., Potter, C.: Discovery of Patterns of Earth Science Data Using Data Mining. In: Next Generation of Data Mining Applications (2004)

Service, Location

► Location-Based Services: Practices and Products

Service Oriented Architecture

► Smallworld Software Suite

Service-Oriented Architecture

ILYA ZASLAVSKY
San Diego Supercomputer Center,
University of California San Diego, San Diego, CA, USA

Synonyms

Organization of IT infrastructure based on services

Definition

Service-oriented architecture (SOA) is an approach to the architecture of information systems based on independently managed loosely-coupled services that can be discovered and invoked, regardless of their underlying implementation, via a relatively small set of well-defined messages passing interfaces. SOA emphasizes separation of services from applications, which are built by combining the services. In this sense it is a complement of grid approaches, which focus on virtualization of resources and mapping of resources to specific implementations.

Main Text

SOA has emerged as the leading approach to cyberinfrastructure development in several large earth sciences projects aimed at creating a domain environment that integrates databases, research tools and software codes developed on multiple platforms and maintained at distributed locations. In such a system, various online and desktop clients can access the same set of core services. For example, within the Consortium of Universities for the Advancement of Hydrologic Sciences (CUAHSI) hydrologic information system (HIS) project (www.cuahsi.org/his), both desktop (ArcGIS, Excel, Matlab, etc.) and online (ArcGIS server-based) client applications can access hydrologic time series from USGS, EPA and other repositories, using the same set of core web services with uniform signatures. Within the Geosciences Network (GEON) (www.geongrid.org), web services are developed to enable ontology-based search against distributed spatial data collections, spatial data registration, and grid node monitoring. While SOA may be implemented using different technologies (SOAP/WSDL services as in the two mentioned projects, REST web services, RPC, DCOM, etc.) the focus is on defining standard and easy to use communication interfaces. The Open Geospatial Consortium (http://www.opengeospatial.org) has focused on defining such interface specifications to support interoperation of geospatial information.

Cross References

- ► Cyberinfrastructure for Spatial Data Integration
- ► Grid
- ► Spatial Information Mediation
- ► Web Services

Service-Oriented Architecture Building Blocks

► Web Services

Services

► Information Services, Geography

Services, Geographic Information

► Web Services, Geospatial

Services, Location-Based

► Web Services, Geospatial

Services, Web

► Location-Based Services: Practices and Products

Shape Function

► Constraint Databases and Data Interpolation

Shifting Geometries

► Positional Accuracy Improvement (PAI)

Shortest Paths

► Fastest-Path Computation

Significance Measures

► Co-location Patterns, Interestingness Measures

Similarity Discovery

► Geospatial Semantic Integration

Similarity Representation

► Geospatial Semantic Integration

Similarity Search

► Indexing and Mining Time Series Data

Similarity, Semantic

► Geospatial Semantic Integration

Simple Features Model

► PostGIS

Simple Object Access Protocol Services

▶ Web Services

Simulated Annealing

▶ Routing Vehicles, Algorithms

Simultaneous Autoregression

▶ Spatial and Geographically Weighted Regression

Simultaneous Autoregressive Model (SAR)

▶ Spatial Regression Models

Simultaneous Spatial Operations

▶ Concurrency Control for Spatial Access
▶ Concurrency Control for Spatial Access Method

Sincerity

▶ Participatory Planning and GIS

Single Image

▶ Photogrammetric Methods

Skyline Queries

NILU THAKUR
Department of Computer Science and Engineering,
University of Minnesota, Minneapolis, MN, USA

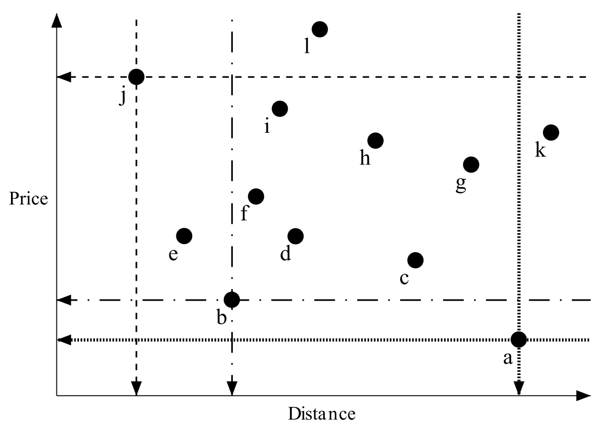
Synonyms

Top-N queries; Search, multi-criteria; Search, multi-dimensional; Ranking Results; Ranking, multi-dimensional; Information Retrieval; Nested-loop, blocked; Divide and conquer; Nearest Neighbor algorithm; Branch and Bound

Definition

Given a set of points $p_1, p_2, \ldots, p_n$, the skyline query returns a set of points P, called skyline points, such that any point $p_i \in P$ is *not dominated* by any other point in the dataset. A point p_1 dominates another point p_2 if p_1 is not worse than p_2 in all dimensions, and p_1 is better than p_2 in at least one dimension. In other words, a point p_1 dominates another point p_2 if and only if the coordinate of p_1 on any axis is smaller than the corresponding coordinate of p_2.

For example, Fig. 1 shows a dataset containing information about hotels. The distance of the hotel from the beach and the price for each hotel are assigned to the X, Y axis of the plot respectively. There are 12 hotels in total represented as data-points in the figure. The interesting data-points in the dataset are $\{a, b, j\}$, because 'a' has the lowest price and 'j' has the least distance from the beach. Although 'b' has neither the shortest distance nor the minimum price, it has a lesser distance value than 'a' and a lower price value than 'j'; hence, 'b' is not dominated by either 'a' or 'j'. All the other points in the dataset are dominated by the set of points $\{a, b, j\}$, i. e., for all other points, both the distance and price value are greater than one or more set of points $\{a, b, j\}$, also called skyline points.



Skyline Queries, Figure 1 Datasets of hotels with their price and least distances from any beach

Historical Background

Skyline and operator queries were first introduced by Bäorzsaonyi et al. [1]. Skyline, in the literal sense, defines an outline, perhaps of buildings or a mountain range against the background of the sky. As the skyline picture of a place shows only the taller or the prominent buildings and structures, similarly, the skyline query on a set of data points returns only the most interesting data points. Skyline queries and skyline operators were introduced to the database context by applying the problem of finding the maxima of a set of points.

In database management systems, rank-aware queries, like top-N or skyline, are often used to mine for the most interesting data items among a large amount of data. Recently, due to the application of skyline queries in multi-criteria decision-making and mobile service applications, skyline queries have gained popularity and are now widely studied in database literature.

Scientific Fundamentals

Algorithms

There are many algorithms for evaluating skyline queries and many more new algorithms have been recently added. A few of the important ones are explained here.

Block Nested Loop (BNL) Algorithm: BNL, proposed in [1], is based on a very straightforward approach of computing the skyline in which each point *p* is compared with every other point in the dataset; if *p* is not dominated, then it becomes a part of the skyline or otherwise rejected. BNL builds on this concept by scanning the data file and keeping a list of candidate skyline points in the main memory. The following list summarizes the steps.

- The first data point is inserted into the list of skyline points.
- All the data points in the data file are scanned and tested for dominance criteria. For each subsequent point *p*, there can be three cases:
  a) If *p* is dominated by any point in the list, it is discarded as it is not part of the skyline.
  b) If *p* dominates any point in the list, it is inserted into the list and all points in the list dominated by *p* are dropped.
  c) If *p* is neither dominated nor dominates any point in the list, it is inserted into the list as it may be part of the Skyline.
- The active list of potential skyline points seen thus far is maintained and each visited point is compared with all elements in the list.
- The list self-organizes because any point found to be dominating other points is moved to the top of the list. This reduces the number of possible comparisons since points that dominate multiple other points are likely to be checked first in order to determine whether a new point is part of the skyline or not.
- This is continued until all the points in the data set get scanned.
- BNL does not require a precomputed index, and the execution remains independent of the dimensionality of the space.

A problem of BNL is that the list may become larger than the main memory. In such cases, all points falling in case 'c' are added to a temporary file. This fact requires multiple passes of BNL. In particular, after the algorithm finishes scanning the data file, only points that were inserted in the list before the creation of the temporary file are guaranteed to be in the skyline and are thus output. The remaining points must be compared against the ones in the temporary file. Thus, BNL has to be executed again, this time using the temporary (instead of the data) file as input. The advantage of BNL is its wide applicability since it can be used for any dimensionality without indexing or sorting the data file.

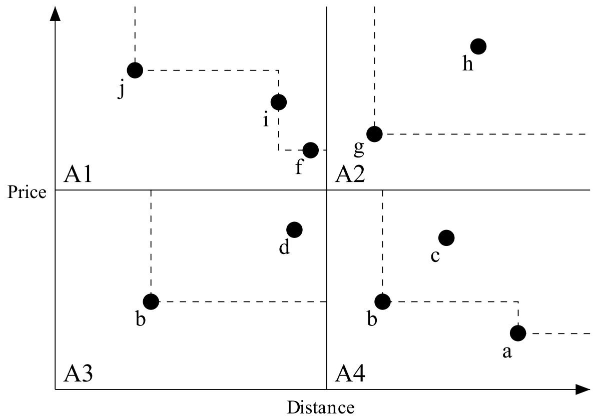
Shortcomings of the BNL algorithm:

- The reliance on main memory, since small memory may lead to numerous iterations
- Its inadequacy for on-line processing, since it has to read the entire data file before it returns the first skyline point
- It does perform redundant work and there is no provision for early termination.

Divide-and-conquer algorithm: This algorithm is based on a very simple approach of incremental evaluation of partial skyline points. This approach divides the large data set in to several smaller partitions which can fit in main memory and then computes a partial skyline of points in every partition using a main memory algorithm [2,3]. The steps are summarized here:

- Recursively large datasets are broken into smaller partitions. This process continues till each smaller partition of the dataset fits in the main memory
- The partial skyline is computed for each partition using any in-memory approach and later these partial skyline points are combined to form the final skyline query.

Figure 2 illustrates how the divide and conquer algorithm works. The data space has a dataset with 10 data points. The data space is further divided into 4 partitions, A_1, A_2, A_3 and A_4, and a partial skyline is computed for these partitions. The data space has its partial skyline as $\{j, i, f\}$, $\{g\}$, $\{e\}$ and $\{b, a\}$, respectively. To obtain the final skyline, all partial skylines from different data spaces need to be compared with other partial skylines for the dominance criteria. As can be seen in the Fig. 2, the partial skyline of data space A_3 must appear in the final skyline, while those in data space A_2 should be discarded immediately as they are dominated by point in data space A_3. Then, each skyline point in A_1 must be compared only with skyline points in A_3, because no point in A_2 and A_4 can dominate points in A_1. In this figure, point 'i' and 'f' from data space A_1 will be removed because they are dominated by data point 'e'. Also, the skyline of data space A_4 should be compared with points in A_3, resulting in the removal of data point 'b'.



Skyline Queries, Figure 2 Divide and conquer approach

Finally, the skyline set will contain the data points {j, e, a} as the final skyline points.

Shortcomings of the Divide and conquer approach are:

- Efficient only for small data sets, because for large data sets partitioning requires reading and writing the entire data set once, and incurs significant IO cost
- Not suitable for online processing, as the output cannot be sent until the partitioning phase is completed.

Nearest Neighbor (NN) Algorithm: The NN algorithm is based on the results of a nearest neighbor search to partition the data universe recursively. It assumes that a spatial index structure on the data points is available for use. In general, NN identifies skyline points by repeated application of a nearest neighbor search technique on the data points using a suitably defined L1 distance norm. The various steps can be summarized as:

- First a nearest neighbor query is performed using an existing algorithm on the R-tree such as [4,5] and the point with minimum distance from the beginning of the index is found. For example, in Fig. 3a, the nearest neighbor in the data-point is 'a', as it is closest to the origin when an L1 distance measure is assumed
- **'a'** divides the entire space into 2d non-disjoint region, which now must be recursively searched for more skyline points
- Regions 2 and 4 need not be searched, because all the points in region 2 are dominated by 'a'. The rest of the regions need to be searched
- Next, the search is recursively done on region 1. In that region, the nearest neighbor would be 'f', then the region is exploded to form additional region. Now with f as the nearest neighbor, region 4 need not be searched and region 4 is added to the pruned region
- In this way, the set of partitions resulting after finding a skyline point are inserted in a to-do list until the to-do list is no longer empty, NN removes one of the partitions from the list and recursively repeats the process
- In the case of high-dimensional data sets, the number of unexplored region grows rapidly and therefore, the non disjoint condition can be relaxed
- The restriction that regions are non-overlapping is relaxed. An assumption can be made that a point query splits each dimension into two regions; instead of exploding a region to 2 to the power of d, it can be reduced to $2d$
- This leads to a fewer number of regions to search at the expense of performing the task of duplicate removal
- Laissez-Faire and propagate are the two duplicate removal techniques that can be used.

Duplicate Removal in the NN algorithm:
Kossmann et al. [6] proposed the following elimination methods for duplicate removal:

Laissez-Faire: This method maintains an in memory hash table which keys in each point and flags it as a duplicate if it is already present in the hash table.

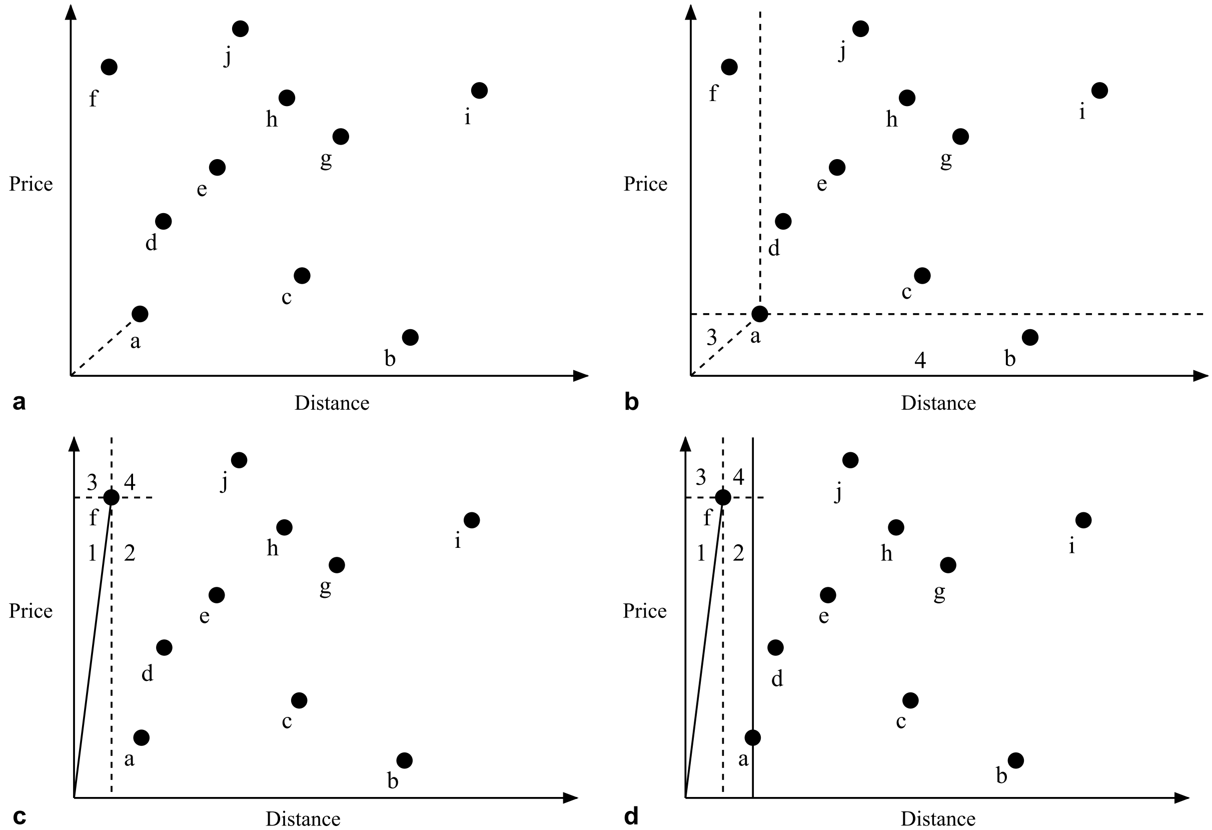
Propagate: When a point p is found as the skyline point, then all the instances of p are removed from all unvisited nodes.

Shortcoming of the Nearest Neighbor algorithm:

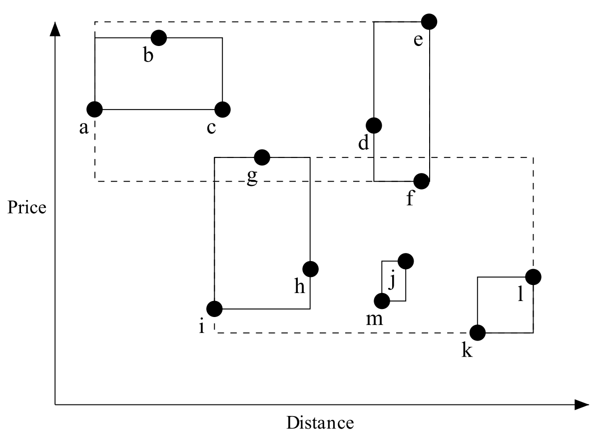
- In the case of high-dimensional data sets, the unexplored region or to-do list grows rapidly and in such cases, the non-disjoint condition needs to be relaxed for high-dimensional datasets.

Branch and Bound Skyline (BBS) Algorithm: The BBS algorithm proposed in [7], like the NN algorithm, is based on nearest neighbor search. Specifically, BBS uses a branch and bound paradigm. For the data set shown in Fig. 4, the steps of the BBS algorithm can be summarized as follows:

- Assuming an R-tree on the data set, the algorithm starts from the root node of the R tree and inserts all its entries in a heap sorted according to their mindist ordering relative to the origin using L1 distance norm
- Then, the entry with the minimum mindist is expanded, which is e7 in this case
- The expansion process removes the entry e7 from the heap and inserts its children e3, e4 and e5
- The next entry to be expanded is e3 with minimum mindist and the data point 'i' is found as the first nearest neighbor. This 'i' becomes part of the skyline
- Then, e6 is expanded and its children are checked for dominance criteria. Only the ones that are not dominated by some points in the skyline are inserted into the heap. For example, in the figure, 'e2' and 'h' are pruned because they are dominated by point 'i'
- The algorithm proceeds in this manner until the heap becomes empty.



Skyline Queries, Figure 3 **a** The nearest neighbor is 'a'. **b** 'a' divides the space in to 2d non-disjoint regions. **c** A search is done on region 1 and point 'f' found. **d** Region 4 is added to the pruned region and need not be searched



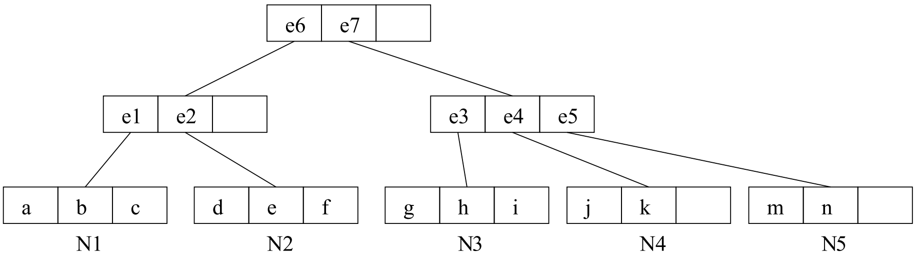
Skyline Queries, Figure 4 Dataset with an R-tree to illustrate the BBS algorithm

Variations of Skyline Queries

Ranked Skyline Queries: Given a set of points in d-dimensional space, a ranked or top-k skyline query takes a parameter 'k' and a preference function 'f', which is monotone on each attribute, and returns the skyline points that have a minimum score according to the input function. In the case of ranked skyline queries, in addition to the distinct score functions used for skyline computation a "total" score function is also provided. Then, all skyline points are returned in order of the "total" score function. An alternate preference function is used instead of the minimum criterion. The priority queue uses the alternate preference-function to compute the Minimum Distance to the elements in the queue.

For example, in Fig. 4, if x and y coordinate of 'i' and 'k' are (3,2) and (9,1), respectively, and if $K = 2$ and the preference function is $f(x, y) = x + 3y^2$, then the output skyline point would be $\{k,12\}$ and $\{i,15\}$. The Branch and bound algorithm can easily handle ranked skyline queries by modifying the mindist definition to reflect the preference function. By contrast, other algorithms like Block nested loop and divide and conquer first try to retrieve the entire skyline, and then sort the skyline according to their score, then outputing the best k points.

Constrained Skyline Queries: A constrained skyline query returns the most interesting points in the data space defined by the constraints. Generally, each constraint is expressed as a range along a dimension and the con-



Skyline Queries, Figure 5 R-tree representation of the Fig. 4 data set

junction of all constraints forms a hyper-rectangle in the d-dimensional attribute space. An example of a constraint skyline query would be a case where a user is interested only in hotels whose nightly rate is in the range of \$40–\$70.

The Branch and Bound skyline algorithm (BBS) can easily process constraint skyline queries by pruning the entries not intersecting the constraint region. In addition to the usual skyline input parameters, the data space is constrained and the skyline queries return skyline points only from the constrained data space. This is typically done by specifying a hyper-rectangle in which all data items have to be located. When inserting objects into the priority queue, objects that lie completely outside the constraint region are pruned.

Dynamic Skyline Queries: A dynamic skyline query specifies an 'm' dimension function $f_1, f_2, \ldots, f_m$, such that each function f_i $(1 < i < m)$ takes as parameters the coordinates of the data points along a subset of d axes. These queries return the skyline in the new data space with dimensions defined by $f_1, f_2, \ldots, f_m$. For example, suppose a data base stores for each hotel its X and Y coordinates and its price i.e, the database contains three dimensions and suppose a user specifies his or her current location (l_x, l_y) and wants to know the most interesting hotels with respect to Euclidean distance and the price of the hotel. In this case, each point P with coordinate (p_x, p_y, p_z) in the original 3D space is transformed to a point p' in the 2D space with coordinates $(f_1(p_x, p_y), f_2(p_z))$ where the dimension functions f_1, f_2 are defined as:

$$f_1(p_x, p_y) = (p_x - u_x)2 + (p_y - u_y)2 \text{ and } f_2(p_z) = p_z \,.$$

Enumerating Skyline Queries: For each skyline point 'p' in the data set, the number of points dominated by 'p' are found by enumerating the queries. This information may be relevant in applications where the goodness of a skyline point is determined by how many other point it dominates. This is done by defining the spatial bound for the region where a skyline point dominates. Then, all points in the data sets are scanned and checked against the spatial extent for each of the skyline points. The total number of point-region intersection gives the required count for all skyline points.

K-dominant Skyline queries: A k-dominant skyline contains all the points that cannot be k-dominated by any other points. A point 'p' is said to k-dominate another point 'q' if there are k dimensions in which 'p' is better than or equal to 'q' and better than 'q' in at least one dimension. For example, in Table 1, $\{p_1, p_2, p_3, p_5\}$ form a free (6-dominant) skyline, $\{p_1, p_2, p_3\}$ form a 5-dominant skyline, while $\{p_1, p_2\}$ form a 4-dominant skyline. p_4 is not a part of the 5-dominant skyline as it is dominated by other points in all dimensions. p_5 is not a part of the 5-dominant skyline as it dominated by other points in all 5-dimensions not involving d4.

Skyline Queries, Table 1 An example data set showing k-dominance

	d1	d2	d3	d4	d5	d6
P1	2	2	2	4	4	4
P2	4	4	4	2	2	2
P3	3	3	3	5	3	3
P4	4	4	4	3	3	3
P5	5	5	5	1	5	5

Spatial Skyline queries: Given a set of data points P and a set of query points Q, each data point has a number of derived spatial attributes, each of which is the point's distance to a query point. An SSQ retrieves those points of P which are not dominated by any other point in P, considering their derived spatial attributes. The main difference with the regular skyline query is that this spatial domination depends on the location of the query points Q. SSQ has application in several domains such as emergency response and online maps [3].

Key Applications

Skyline queries and their variations are used in many applications involving multi-criteria decision-making, mobile applications, streaming applications, spatial applications and others.

Multi-criteria Decision Making

This area can again be divided into applications with static data points and those with dynamic data points. Examples with static data points are cell phone finder and movie rating, whereas location based services with moving objects are examples of decision-making with dynamic data.

Cell Phone Finder

A person looking to buy a cell phone at a website may care about a large number of features like weight, size, talk time, standby time, screen size, screen resolution, camera quality, color, brand and other things specific to his/her choice. There will be too many phones for a person to examine manually. In such cases, computing the skyline over the desired cell phone features may remove a large number of cell phones whose features are not better than those in the skyline, thus leaving a manageable number of phones for evaluation.

Movie Rating

A person looking for top ranked movies based on the ranking given by other users on a particular movie website might be interested in only top ranking movies. In such cases, the rating of each user corresponds to a dimension in the data set, and due to a large number of users, the data set obviously becomes a high-dimensional one. The skyline of the data set will contain only top rated movies while the low ranked queries will be ignored and not become part of the skyline. Note that in both the above cases, ranking can be done by providing some preference functions and requesting users to provide some weight assignments for their preferred features or more trusted users in the latter case. However, providing such weight assignments for a large number of dimensions is not always easy without any initial knowledge about the data. For example, it is not clear how weight assignments can be provided to aggregate the talk time and camera quality of a phone into one grade. Computing skylines in high-dimensional data sets is challenging because of the large number of skyline points. On the movie ranking website, for example, it is nearly impossible to find a movie which is ranked lower than another movie by all the users. Such a blowup in the answer set not only renders the skyline operator worthless (with respect to the desired pruning of candidates), but it also results in high computational complexity for both index and non-index methods as many pairwise comparisons are performed without affecting any pruning.

Location Aware Mobile Services

Suppose that a multidisciplinary task force team located at different and fixed offices wants to put together a list of restaurants for their weekly lunch meetings. These meeting locations must be interesting in terms of traveling distances for all the team members; for each restaurant *r* in the list, no other restaurant is closer to all members than r. Also, considering that the team members are mobile and change location over time, the computation of skyline becomes complex. In this case, the restaurants distance attributes based on which the domination is defined are dynamically calculated based on the user's query or based on the location of the team members.

Road Network Services

A person looking for a gas station may prefer to minimize detours rather than distance, while a user searching for an emergency room is likely to be interested in minimizing the distance and is insensitive to the detour. In such cases, the skyline operator should return larger result sets from which the user can choose the route. The skyline mechanism returns a result if no other result exists that is better with respect to all the criteria considered. This is useful when a total ordering cannot be defined on the space of all criteria.

Future Directions

Progress is on-going with regards to work in skyline computations and important avenues of research in the future include:

- Alternative optimal and progressive algorithms for high-dimensional spaces where R trees are inefficient
- Fast retrieval of approximate skyline points, i.e, points that do not belong to a skyline but are very close
- Challenges and application of spatial skyline queries in metric spaces such as road networks.

S

Cross References

► Aggregation Query, Spatial
► R-Trees – A Dynamic Index Structure for Spatial Searching
► Vector Data

Recommended Reading

1. Baorzsaonyi, S., Kossmann, D., Stocker, K.: The Skyline Operator. ICDE (2001) pp. 421–430, Heidelberg, Germany
2. Stojmenovic, I., Miyakawa, M.: An optimal parallel algorithm for solving the Maximal Elements Problem in the Plane. Parallel Computing (1988) **7**(2):249–251
3. Matousek, J.: Computing dominance in E^n. Information processing letters, Vol. 38 pp. 277–278 (1991)
4. Roussopoulos, N., Kelley, S., Vincent, F.: Nearest Neighbour Queries. SIGMOD (1995)
5. Hjaltason, G., Samet, H.: Distance Browsing in Spatial Databases. ACM TODS **24**(2):265–318 (1999)

6. Kossmann, D., Ramsak, F., Rost, S.: Shooting Stars in the Sky: An Online algorithm for Skyline Queries. In: Proceedings of VLDB'02, pp. 275–286 Hong Kong, China (2002)
7. Papadias, D., Tao, Y., Fu, G., Seeger, B.: An optimal and progressive algorithm for Skyline Queries. In: ACM SIGMOD, San Diego, CA, USA (2003)
8. Chomicki, J., Godfrey, P., Gryz, J., Liang, D.: Skyline with Presorting. In: proceedings of ICDE'03, IEEE Computer Society, pp. 717–816 Bangalore, India (2003)
9. Lin, X., Yuan, Y., Wang, W., Lu, H.: Stabbing the Sky: Efficient Skyline Computation over Sliding Windows. In: Proceedings of ICDE'05, IEEE Computer Society, pp. 502–513 Tokyo, Japan (2005)
10. Papadias, D., Tao, Y., Fu, G., Seeger, B.: Progressive Skyline Computation in Database Systems. ACM Trans. Database Syst. **30**(1),41–82 (2005)
11. Tan, K.L., Eng, P.K., Ooi, B.C.: Efficient Progressive Skyline Computation. In: Proceedings of VLDB'01, pp. 301–310 (2001)

Smallworld GIS

► Smallworld Software Suite

Smallworld Software Suite

ROBERT P. LAUDATI
Marketing Program Manager, GE Energy,
Englewood, CO, USA

Synonyms

Smallworld GIS; GE smallworld; Software; Object-oriented; Components; Reuse; Integration; Database management; Manifolds; Manifold rules; Version managed datastore; Version management; Versioned B-tree; B-tree, versioned; Locking; Check-out; Conflict resolution; Distributed databases; Multiple worlds; Worlds, multiple; Magik, smallworld; Service oriented architecture; OGC standards, ODBC; COM/OLE; Oracle spatial

Definition

Smallworld Core Spatial technology is an object-oriented, database-driven product that provides a powerful, consistent architecture for geospatial applications such as those used for planning electric, gas and water distribution systems, designing telecommunications networks and evaluating strategic market opportunities. Encapsulated, reusable components enable rapid development with reduced custom code, driving down maintenance and upgrade costs-build once, use many times. Smallworld technology has been proven to deliver solutions for thousands of users managing terabytes of data across complex, distributed operations. The software integrates with other products that require spatial information, including systems for customer relationship management, market analysis, network and work management.

Historical Background

Smallworld Systems Limited was founded in 1988 by ten pioneers with extensive experience in computer graphics and large databases obtained during careers at the CAD-Centre in Cambridge, England, and Cambridge Interactive Systems (CIS), the producer of Medusa, which became the leading CAD product in Europe.

During its first year, Smallworld was commissioned by the world's leading technology vendors to research and assess the needs of the GIS marketplace in Europe, with a focus on utility companies. More than 650 organizations in ten countries were interviewed, including transportation, cartographic, and government agencies. This wealth of information provided Smallworld with revealing insights into the real needs of the marketplace and an in-depth understanding of the GIS user community.

In April 1990, the Smallworld GIS product was first launched at the European GIS conference (EGIS) in Amsterdam, and by August 1990, the company had signed its first customer. In 1991, sales and support offices were opened in the Netherlands and Sweden, and the company had installed systems in utility companies in the United Kingdom, Germany, Switzerland, the Netherlands, and Sweden. By the end of 1992, Smallworld Systems had over 60 customers in Europe.

Smallworld technology was introduced to the North American market in 1993 at the Automated Mapping/Facilities Management (AM/FM) show in Orlando, Florida. Soon after, Smallworld established a U.S. subsidiary in Colorado. By July of 1993, Smallworld had its first success in North America, winning a contract with Centra Gas in Manitoba, Canada. This was quickly followed by contracts with Public Service Company of Colorado, Entergy Services Corporation, Commonwealth Edison, and Wisconsin Gas.

In October of 2000, Smallworld Systems became a business unit of GE Energy Management Services, a part of GE Energy. GE Energy (http://www.geenergy.com) is one of the world's leading suppliers of power technology, energy services and information management systems with 2005 revenues of $16.5 billion, serving customers in over 100 countries for over a century.

Scientific Fundamentals

Smallworld Core Spatial Technology ("Core") is based on database management foundations rather than file or CAD foundations [1,2]. The underlying principle is that the system manages a single logical database that manages both spatial and non-spatial attributes together in a seamless, "object-centered" approach. One of the benefits of this methodology is that the spatial data can be modeled as it really is – a seamless carpet of topologically structured geometry.

Database Objects and Topology

Smallworld Core models the world as a collection of objects that have relationships and attributes. Complex object relationships are modeled by one-to-one, one-to-many or many-to-many relations. Object attributes are non-spatial (represented as numbers or characters for example) and spatial (represented as points, chains and areas). A non-spatial attribute may be defined as a 'logical' attribute; such an attribute is not defined in the database, it is derived by a method on the object when needed.

Spatial objects can contain three primary types of geometry attributes – area, chain or point. Each type of geometry can be either topological or simple. Topological geometry can share components with other topological objects (the nodes, links and polygons define it), while simple geometry has a visual representation, but cannot share components. Topological relationships are modeled by points, chains, and areas sharing nodes, links or polygons. A chain consists of a linear sequence of links (x,y or x,y,z coordinate arrays) and an area consists of a contiguous set of polygons. The two level structure allows for a much richer modeling of topology than a layer based approach. One object may have multiple geometries and many objects may share one geometry, as shown below:

Manifolds are used to group together those geometric attributes in the system whose geometry may interact. This enables Smallworld to answer queries about shortest path, equipment on pipes or cables, adjacent areas, etc. The ways in which a new geometry in a manifold interacts are determined by a set of manifold rules. A manifold rule describes what happens to the underlying connective objects when two topological geometry objects coincide. It has two parts:

- A geometric part defining the geometric objects to which it applies
- A connection rule describing what happens to the connective objects

A manifold rule that involves a specified object attribute is an *explicit manifold rule*; manifold rules involving all geometry objects of a given type are *default manifold rules*. Each time a geometric attribute of an object is inserted, it refers to its manifold rules and creates the appropriate topological structure for its geometry. The default rule should depend upon the modeling requirements of the network being modeled.

Version Managed Datastore

Smallworld's database architecture includes a Version Managed Datastore (VMDS), which is a logical view of a physical database that may share data with other views or alternatives. Version management is a method of handling long transactions by giving the database the capability to store multiple, possibly conflicting, changes in the database. As a result, many different users updating a VMDS database can gain access to the entire database subject to their own modifications. Each user has a personal view of the database through an alternative, without having to replicate data common to other alternatives.

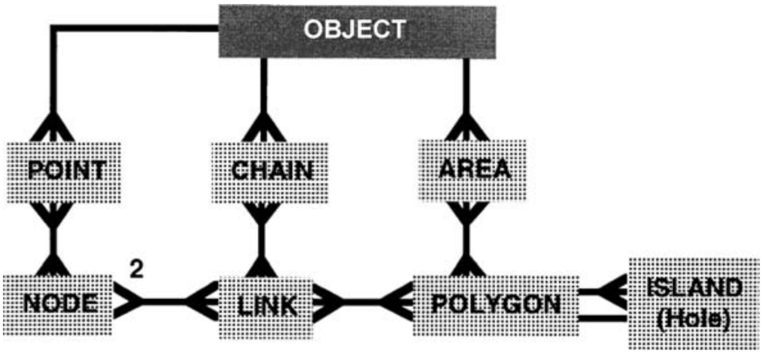
It is common practice that at the lower levels of a data management system, the mapping between logical and physical blocks is handled by means of a page map. Different versions of the database are represented by different page maps that may well share many of the actual physical disk blocks. However, a single level page map for an application with the data volumes typical of a geospatial system is likely to be extremely large (certainly too large to hold in main memory).

The approach for version management in Smallworld Core is based on handling versions within the fundamental storage mechanisms of the system with what is known as a versioned B-Tree, where the data is held as a balanced hierarchy of indices and the leaf blocks hold the actual data. The non-leaf nodes are equivalent to a hierarchically structured page map. With B-Trees, the fundamental versioning mechanism is built into the underlying data structure itself. This has the properties of low space overhead and almost zero performance overhead during conventional access. This level of the implementation permits multiple versions of the data to co-exist, with blocks that are common to two or more versions being shared between them. Note that the approach does not demand that all blocks reside in the same file. In particular, benefit can be gained in the long term by storing a version on a mass storage medium such as CD ROM and maintaining the changes on conventional disk.

In order to determine the differences between a version and one of its ancestors, it is only necessary to examine those blocks in that version that differ from the ancestor's version, rather than all the blocks in the database.

Different versions of the same tree are simply accessed via their different root blocks, and the action of committing a new version to disk simply involves storing the new root block number in an appropriate place. The old version con-

S



Smallworld Software Suite, Figure 1 Smallworld Geometry Structure

tinues to remain accessible while its root block number is still known.

The basic mechanism is described in [2] and offers a series of significant benefits:

- There is no need to go through a check-out process before an update commences
- Updates may be freely mixed with queries across any part or the whole of the database
- Panning and zooming across the whole database may be freely mixed with updates
- It provides a mechanism to hold design alternatives without replicating common data

The many versions present in an evolving version managed datastore form a hierarchy, the root of which is probably sometimes considered a sort of master version, to which ultimately changes in descendant versions will be posted. Some systems (e. g. CAD systems) already support a simple form of version management for single users, in allowing a series of checkpoints to be stored in one design file. This is not so much a hierarchy as a linear sequence of versions, held without replicating data that has not changed between checkpoints.

Conflict Resolution

In a normal database transaction model, there are two approaches adopted for avoiding conflict. One is a system of a *priori* locking (pessimistic locking), which guarantees that no user can update the database and cause a conflict, because he is explicitly prevented from doing so by the locks. Alternatively, a more modern approach is to use optimistic locking, which may cause an attempted update to be lost at the time a conflicting change is committed. In a normal commercial application, the work involved in making the change is small, and so it does not matter to redo it.

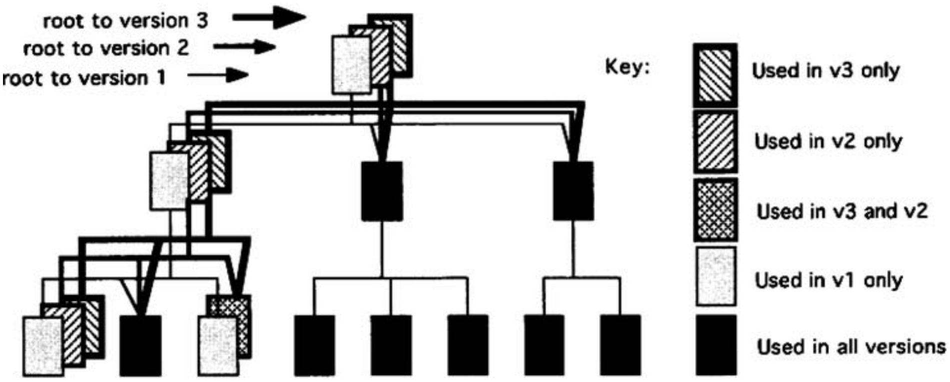
A system of check-out with locking is similar in nature to pessimistic locking, which is constraining in a system with long transactions. Check-out without locking means either all work may be lost when conflict is detected, or a system of conflict resolution is required. Most systems that use check-out take the pessimistic route, avoiding conflict at all costs.

In Smallworld's version managed database, however, the system can be tuned by means of semantic locks to behave in any way between the extremes of pessimism and optimism. Thus one way of resolving conflict is to prevent it in the first place, e. g. by constraining each user to a particular non-overlapping geometric area, or a constraint on the kind of data that can be updated, or maybe even a combination of the two.

However, in a real system, it is impractical and undesirable to avoid conflict at all times, and therefore it is necessary to be able to detect conflicts and deal with them when they occur. Conflict resolution in Smallworld Core occurs as part of the merge operation, when changes that have been posted to a parent alternative are merged down into a descendant alternative. The result of this operation is that in some cases, conflicting data may be removed to a neutral place in the merged alternative, giving the user the option of piecing together the data interactively.

Distributed Databases

In the markets that Smallworld technology serves, there is often the need to distribute the data and applications to remote site with poor connectivity, since the lack of network bandwidth often precludes the implementation of a single database that can serve all locations with adequate performance. As a result, one of the key capabilities of Smallworld Core is a set of data distribution capabilities. In implementing a distributed database, there is always a trade-off between holding each item of data at one location only and replicating all data at every location. In the first case, the problem of optimizing distributed queries is considerable; fast and expensive communications are essential, and systems at all locations may become unusable if one location suffers a hardware failure. The use of



Smallworld Software Suite, Figure 2 Illustration of block sharing in a vision managed B*-Tree

data replication can considerably speed data access and provide robustness against remote location failure, but poses potential problems in maintaining the consistency of data that is duplicated at different locations [3,4].

Smallworld's Versioned Managed Datastore provides a better solution to this problem than conventional DBMS's due to its unique version management capabilities. A single master version, which is updated very infrequently, is replicated at every location. Users at any one location will, in the majority of cases, make database modifications that do not conflict with those made elsewhere. For example, users at a regional office of a water utility might modify burst pipe information, but only for water mains in a particular geographical area. This pattern of use fits well with the idea of having a separate alternative hierarchy at each location, each derived from the common replicated master alternative (root).

The changes held explicitly at each location can be identified and distributed to other locations. The system only requires slow communication links because typically changes are small and can often be transferred in batches rather than immediately.

The actions of dealing with conflicting changes are inevitably more difficult to organize in a distributed environment, but the use of the Version Managed Datastore can ensure that conflicts are detected and that users always see a self-consistent view of the database.

Scalability & Performance

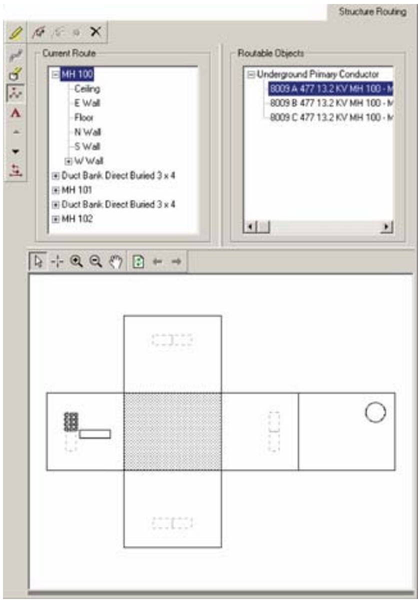
The Version Managed Datastore holds all of the spatial and non-spatial data seamlessly; there are no tiles or layers. The basic storage structure is tabular, but unlike a conventional relational database, the access language is the object-oriented language, Smallworld Magik. Fast spatial retrievals are achieved by means of a modified quadtree indexing technique, which neither complicates nor compromises the logical data structure. The primary key of all spatial data contains an automatically generated 'spatial key'. This method both clusters data spatially on disk and provides an efficient index.

Because most geographic objects are very small compared to the world modeled by the GIS system, in order to avoid trapping objects near the root of the tree, the subdivision method is modified so that each quadtree cell is divided into four parts which overlap slightly, i. e. each quadtree sub-square is slightly more than half the dimension of its parent square. The overlap never needs to be more than the size of the largest object in the database, and in practice can be less than this. The optimum overlap depends on some statistic of the object size, such as the mean object size times a factor. This slight modification to the simple quadtree key lead to worthwhile performance improvements for retrieving objects inside an area and in finding objects surrounding a point compared to the simple quadtree key mechanism.

Multiple Worlds

Although it is essential in many cases to model the real world with a seamless geographic database, there are also cases where some information needs to be handled in a different coordinate system, or what is known in Smallworld terms as a different world. Examples include applications where part of the model is represented as a schematic drawing, either as an alternative representation or as an expanded detail of part of the geographic world such as the internal structure of an electric utility substation. In such cases, although the geometry may be spread across different coordinate systems, it is important to handle the topology so that applications such as network analysis can trace across these multiple worlds seamlessly.

Smallworld Core handles multiple worlds as an extension to the quadtree indexing scheme described above. The spatial index can be extended to include extra high order bits. In this way topological tracing operations work in an identical manner regardless of whether the geometry is in a single coordinate system or multiple coordinate systems.



Smallworld Software Suite, Figure 3 Internal World example in Smallworld Core showing a "butterfly" diagram representing a utility manhole

A further extension of this idea allows the data to be partitioned according to various non-spatial criteria. In Smallworld Core, this technique is used to control both drawing priority and to implement the idea of multiple worlds. A number of high order bits are reserved to store a world ID, and the next block of bits is reserved to store drawing priority.

Smallworld Magik

Smallworld Core is built using an interactive object-oriented programming language called Smallworld Magik. Magik takes the place of both the macro language and systems language in conventional geospatial systems. This language is used for far more than the occasional writing of a macro or for minor customization; most of the Smallworld system is written in Magik, and advanced industry applications are developed in it. The Magik environment is underpinned by a virtual machine, written in C, which contains a full and extensible set of primitives for handling graphics and interaction, database, and remote access to alien processes, such as external database systems and analysis systems.

The Magik language has a syntax which is procedural in style with familiar and readable constructs for conditionals, expressions, loops, and procedure calls. The message expression is formed by a concatenation of the object class and the message. The semantics have primarily been inspired by Smalltalk, Lisp and Clu. In particular the ability to define iterators and iterator methods (borrowed from Clu) is particularly powerful. Magik supports several key aspects of object-oriented systems, including encapsulation and multiple inheritance. Furthermore, in Magik object oriented code can be freely mixed with procedural code, and the language is also weakly typed, providing considerable additional flexibility by run-time message evaluation in a polymorphic system.

The inheritance relationships for the objects in the Magik environment can be complex but inheritance diagrams can be generated automatically. Also, the implementing class of any particular method can be easily identified. The annotated source code for most of the environment is available online. There are code browsers and object inspectors; the system can locate methods and variables given only part of the name. Debugging tools include tracing of calls and access to global variables, and in Magik it is possible to fix and continue after an error.

Interoperability

In the markets that GE Energy serves, companies typically have complex IT environments with heterogeneous systems, and as a result, Smallworld technology has focused on delivering products that support a wide range of interoperability tools to integrate systems throughout the enterprise, including:

- Spatial Object Manager (SOM) – The Smallworld SOM architecture is a powerful virtual database platform for integrating data from other geospatial systems seamlessly into one environment, leveraging information elsewhere in the company as well as from other agencies without having to create redundant copies of data.
- Enterprise Application Integration (EAI) – EAI is a key integration technology for many companies, and the Smallworld platform leverages this "message bus" architecture through a module that supports various messaging systems such as TIBCO, Vitria, and MQ Series, as well as JMS, CORBA, XML input & output and a low-level Java interface.
- Services Oriented Architecture (SOA) – GE Energy has invested heavily in SOA and provides a "geospatial server" capability that extends the Smallworld Internet Application Server (SIAS) lightweight client to act as a true application server for web services that expose Smallworld data and applications.
- Open Geospatial Consortium (OGC) Standards – Smallworld Core supports the Web Mapping Service as both a WMS client and server and continues to add capabilities around OGC compliance including Web Feature Service (WFS) support.
- Oracle, Oracle Spatial, and Oracle Workspace Manager integration tools – GE Energy is working close-

ly with Oracle to develop integration between the two platforms. Smallworld products can seamlessly read and write data residing in an Oracle or Oracle Spatial database through the SOM architecture, and the long transaction environment in Smallworld can be synchronized with Oracle data using an interface to Oracle's Workspace Manager.

- Open Database Connectivity (ODBC) – Smallworld supports open access to data via ODBC. ODBC allows the Smallworld environment to access data from any other ODBC compliant database and be used seamlessly in a Smallworld application. Smallworld SQL Server provides open access via SQL to all Smallworld data for interrogating relational databases and for use by other enterprise applications.
- COM/OLE – Smallworld's COM automation interface allows Smallworld Core to act as an automation client, invoking tasks in other applications and as an automation server to other applications. This capability remains an important integration point for desktop productivity with Microsoft applications such as Xcel, Word, and Outlook.

Key Applications

Although Smallworld technology supports a complete array of geospatial applications, the company's development history and early successes have led to a market focus and leadership position in the electric, gas, and water utility and telecommunications industries. Today, Smallworld technology is deployed in the largest telecommunication and utility companies worldwide due to its powerful combination of geospatial capabilities and advanced management of linear networks. Other industries including transportation, federal and municipal government, and land management have also benefited from Smallworld's performance and scalability to distribute data to large numbers of users.

Smallworld's product offerings include the essential geospatial platform along with applications focused on the electric, gas, water, and telecommunications industries, such as asset management, outage management, engineering design, internet/intranet access, and mobile clients. A partial listing of available Smallworld products include:

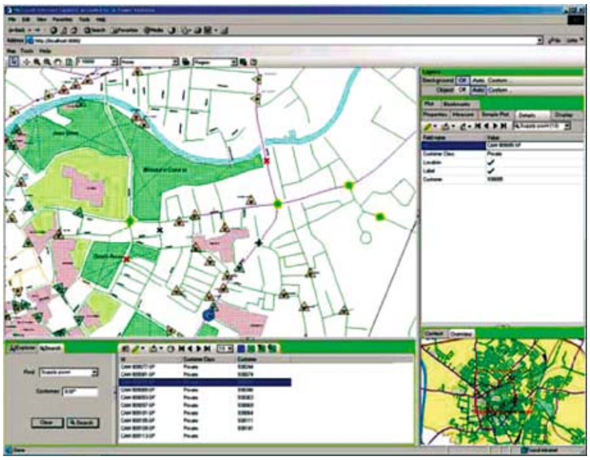
Enabling Geospatial Products	
Smallworld Core Spatial Technology	Core geospatial system
Smallworld Internet Application Server	Internet/Intranet
Smallworld Spatial Intelligence	Desktop analysis client
Smallworld Spatial Object Manager	Virtual geospatial data reader
Smallworld Field Information System	Mobile geospatial application

Utilities	
Smallworld PowerOn	Electric distribution outage management
Smallworld Design Manager	Utility engineering design
Smallworld Analysis & Optimization	Electric power flow analysis
Electric Office	Electric T&D portfolio solution
Gas Distribution Office	Gas distribution portfolio solution
Global Transmission Office	Pipeline transmission portfolio solution
Field Force Automation	Mobile work force management

Telecommunications	
Physical Network Inventory (PNI)	Physical network design and documentation
Logical Network Inventory (LNI)	Logical network design and documentation
Bearer Management	Integration with logical inventory systems

Future Directions

GE Energy is taking the next step in providing complete industry solutions by offering standard Smallworld utility and telecommunications suites. These software suites, or "Offices", address many of the key trends in the utility industry, including the continued incorporation of standard technology such as Oracle Spatial and Java/J2EE as well as the adoption of industry standard interfaces and protocols such as SOA, OGC, CIM, MultiSpeak, and PODS, to name a few [5].



Smallworld Software Suite, Figure 4 Smallworld Internet Application Server

Moreover, Smallworld technology is a key platform for GE Energy's development of their software portfolio to support an "intelligent grid", which consists of an information architecture and infrastructure that enables continuous automated monitoring of a utility's assets. The Smallworld Office provides the underlying spatial asset management system, enterprise business intelligence, asset optimization, and resource optimization tools, and also serves as the data integration portal between other grid monitoring and automation applications.

Cross References

- ► Change Detection
- ► Data Models in Commercial GIS Systems
- ► Indexing, BDual Tree
- ► Market and Infrastructure for Spatial Data
- ► Network GIS Performance
- ► Web Services, Geospatial

Recommended Reading

1. Chance, A., Newell, R.G., Theriault, D.G.: An Object-Oriented GIS – Issues and Solutions, EGIS '90 Conference Proceedings, Amsterdam, April (1990)
2. Easterfield, M.E., Newell, R.G., Theriault, D.G.: Version Management in GIS Applications and Techniques, EGIS '90 Conference Proceedings, Amsterdam, April (1990)
3. Ceri, S., Pelegatti, G.: Distributed Databases: Principles and Systems, McGraw-Hill (1984)
4. Savage, J.F.: Anticipating the Future of AM/FM Systems: The Benefits of Distributed Relational Technology. Proceedings of Executive Management Symposium, AM/FM International, Keystone, Colorado, August (1991)
5. GE Energy: Smallworld Core Spatial Technology. GE Energy Web Site, http://www.gepower.com/prod_serv/products/gis_software/en/core_spatial.htm (2007)

Smart Buildings

► Indoor Positioning

SMIL

► Scalable Vector Graphics (SVG)

SNAP

► Temporal GIS and Applications

Snapshots

► Temporal GIS and Applications

Social Disorganization

► Crime Mapping and Analysis

Software

► Smallworld Software Suite

Software Design Model

► UML: Unified Modeling Language

Soil and Water Assessment Tool "SWAT"

► Rural Hydrologic Decision Support

SOM Usage

► Self Organizing Map (SOM) Usage in LULC Classification

Space-Filling Curves

MOHAMED F. MOKBEL[1], WALID G. AREF[2]
[1] Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA
[2] Department of Computer Science, Purdue University, Indianapolis, IN, USA

Synonyms

Distance-preserving mapping; Locality-preserving mapping; Multi-dimensional mapping; Linearization

Definition

A space-filling curve (SFC) is a way of mapping a multi-dimensional space into a one-dimensional space. It acts like a thread that passes through every cell element (or pixel) in the multi-dimensional space so that every cell is visited exactly once. Thus, a space-filling curve imposes a linear order of points in the multi-dimensional space. A D-dimensional space-filling curve in a space of N cells (pixels) of each dimension consists of $N^D - 1$ segments where each segment connects two consecutive D-dimensional points. There are numerous kinds of space-filling curves (e. g., Hilbert, Peano, and Gray). The difference between such curves is in their way of mapping to the one-dimensional space, i. e., the order that a certain space-filling curve traverses the multi-dimensional space. The quality of a space-filling curve is measured by its ability to preserve the locality (or relative distance) of multi-dimensional points in the mapped one-dimensional space. The main idea is that any two D-dimensional points that are close by in the D-dimensional space should be also close by in the one-dimensional space.

Historical Background

Space-filling curves were discovered by Peano [26] in 1890, where he introduced mapping from the unit interval to the unit square. Hilbert [10] generalized the idea to a mapping of the whole space. Following the Peano and Hilbert curves, many space-filling curves were proposed, e. g., [3,22,28]. Space-filling curves are classified into two categories: recursive space-filling curves (RSFC) and non-recursive space-filling curves. An RSFC is a SFC that can be recursively divided into four square RSFCs of equal size. Examples of RSFCs are the Peano SFC, the Gray SFC, and the Hilbert SFC. For the past two decades, recursive space-filling curves have been considered a natural method for locality-preserving mappings. Recursive space-filling curves are a special case of fractals [15]. Mandelbrot [15], the father of fractals, derived the term *fractal* from the Latin adjective *fractus*. The corresponding Latin verb *frangere* means "to break" or "to fragment". Thus, fractals divide the space into a number of fragments, visiting the fragments in a specific order. Once a fractal starts to visit points from a certain fragment, no other fragment is visited until the current one is completely exhausted. By dealing with one fragment at a time, fractal locality-preserving mapping algorithms perform a local optimization based on the current fragment.

Although space-filling curves were discovered in the last century [10,22,26], their use in computer science applications was not discovered until recently, as it is mainly motivated by the emergence of multi-dimensional applications. In particular, space-filling curves have been used as a mapping scheme that supports spatial join algorithms [4,24], spatial access methods [7,12,13], efficient processing of range queries [3,6,11], nearest-neighbor queries [14], data-parallel applications [25], disk scheduling [18], memory management [16], mesh-indexing [23] and image processing [30].

Numerous algorithms were developed for efficiently generating different space-filling curves that include recursive algorithms for the Hilbert SFC [5,9,29], recursive algorithms for the Peano SFC [29], and the table-driven algorithms for the Peano and Hilbert SFCs [9]. The clustering and mapping properties of various space-filling curves have been extensively studied in the literature (e. g., see [17,20,21]). For a survey of more types of space-filling curves and a comprehensive comparison of various space-filling curves, the reader is referred to [17,20,27].

Scientific Fundamentals

Mapping Scheme

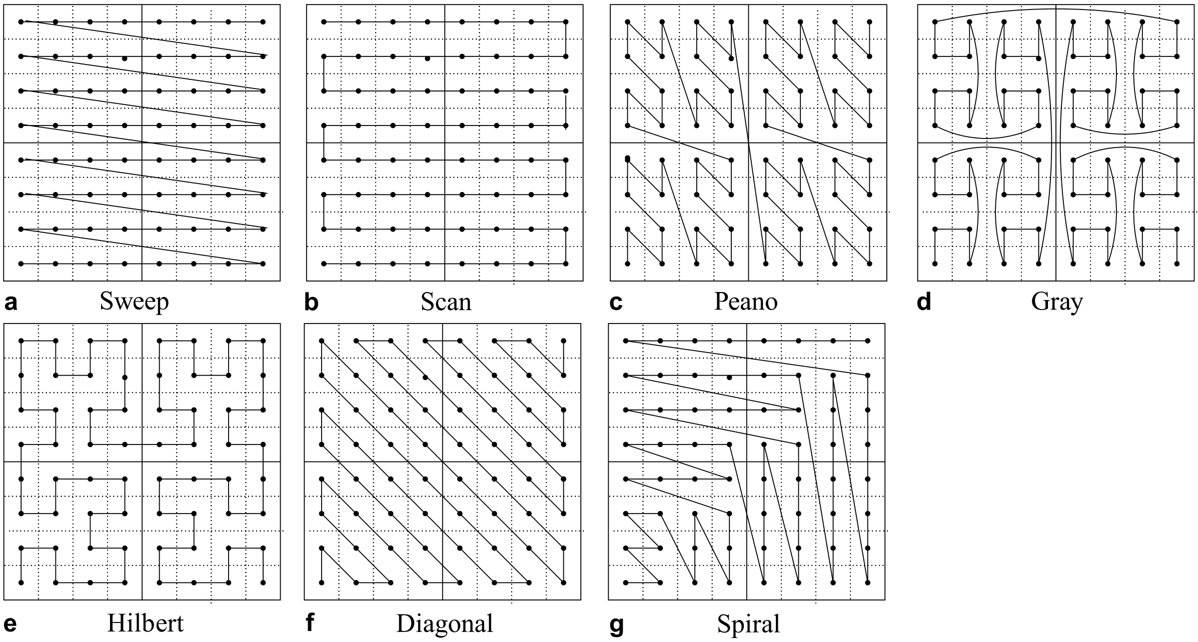
Figures 1 and 2 give examples of two- and three-dimensional space-filling curves with grid sizes (i. e., number of points per dimension) eight and four, respectively. Non-recurisve space-filling curves include the Sweep SFC (Figs. 1a, 2a), the Scan SFC (Figs. 1b, 2b), the Diagonal SFC (Fig. 1f), and the Spiral SFC (Fig. 1g). Recursive space-filling curves include the Peano SFC (Figs. 1c, 2c), the Gray SFC (Figs. 1d, 2d), and the Hilbert SFC (Figs. 1e, 2e). Table 1 gives the first 16 visited points for the Peano, Gray, and Hilbert space-filling curves.

Space-Filling Curves, Table 1 The first 16 traversed points by two-dimensional Peano, Gray, and Hilbert space-filling curves

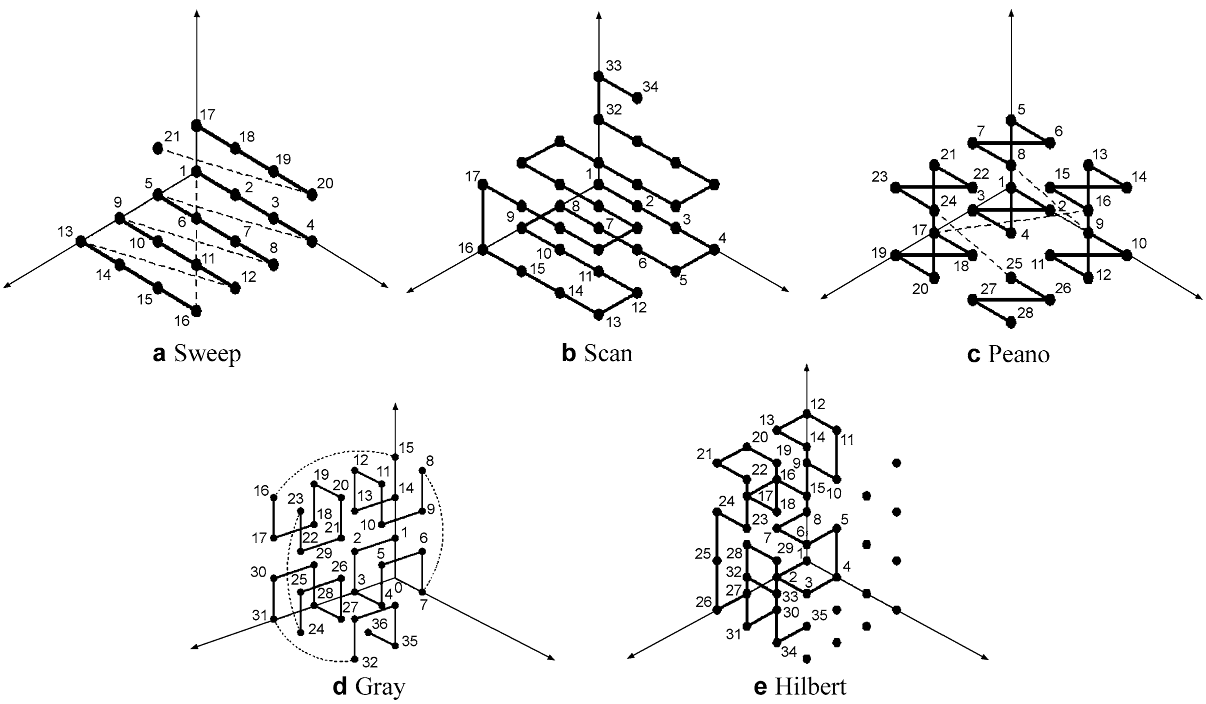
Point	Peano	Gray	Hilbert	Point	Peano	Gray	Hilbert
0	(0,0)	(0,0)	(0,0)	8	(2,0)	(3,3)	(2,2)
1	(0,1)	(0,1)	(0,1)	9	(2,1)	(3,2)	(3,2)
2	(1,0)	(1,1)	(1,1)	10	(3,0)	(2,2)	(3,3)
3	(1,1)	(1,0)	(1,0)	11	(3,1)	(2,3)	(2,3)
4	(0,2)	(1,3)	(2,0)	12	(2,2)	(2,0)	(1,3)
5	(0,3)	(1,2)	(3,0)	13	(2,3)	(2,1)	(1,2)
6	(1,2)	(0,2)	(3,1)	14	(3,2)	(3,1)	(0,2)
7	(1,3)	(0,3)	(2,1)	15	(3,3)	(3,0)	(0,3)

Segment Types

A space-filling curve consists of a set of segments. Each segment connects two consecutive multi-dimensional points. Five different types of segments are distinguished, namely, *Jump*, *Contiguity*, *Reverse*, *Forward*, and *Still*. A *Jump* segment in an SFC is said to happen when the distance along any of the dimensions between two consecutive points in the SFC is greater than one. Similarly, a *Contiguity* segment in an SFC is said to happen when the distance along any of the dimensions between two consecutive points in the SFC is equal to one. On the other side, a segment in an SFC is termed a *Reverse* segment if the projection of its two consecutive points along any of the dimensions results in scanning the dimension in decreasing order. Similarly, a segment in an SFC is termed a *Forward* segment if the projection of its two consecu-



Space-Filling Curves, Figure 1 Two-dimensional Space-Filling Curves



Space-Filling Curves, Figure 2 Three-dimensional Space-Filling Curves

tive points along any of the dimensions results in scanning the dimension in increasing order. Finally, a segment in an SFC is termed a *Still* segment when the distance along any of the dimensions between the segment's two consecutive points in the SFC is equal to zero. Closed formulas to count the number of *Jump*, *Contiguity*, *Reverse*, *Forward*, and *Still* segments along each dimension can be found in [19,20].

Irregularity

An optimal locality-preserving space-filling curve is one that sorts multi-dimensional points in ascending order for all dimensions. However, in reality, when a space-filling curve attempts to sort the points in ascending order according to one dimension, it fails to do the same for the other dimensions. A good space-filling curve for one dimension is not necessarily good for the other dimensions. In order to measure the mapping quality of a space-filling curve, the concept of *irregularity* has been introduced as a measure of goodness for the order imposed by a space-filling curve [17]. Irregularity introduces a quantitative measure that indicates the non-avoidable reverse order imposed by space-filling curves for some or all dimensions. Irregularity is measured for each dimension separately, and gives an indicator of how a space-filling curve is far from optimal. The lower the irregularity, the better the space-filling curve. The irregularity is formally defined as: For any two points, say P_i and P_j, in the D-dimensional space with coordinates $(P_i.u_0, P_i.u_1, \ldots, P_i.u_{D-1})$, $(P_j.u_0, P_j.u_1, \ldots, P_j.u_{D-1})$, respectively, and for a given space-filling curve S, if S visits P_i before P_j, an irregularity occurs between P_i and P_j in dimension k if $P_j.u_k < P_i.u_k$. Closed formulas to count the number of irregularities for various space-filling curves can be found in [17].

Key Applications

Pre-processing for Multi-dimensional Applications: Multimedia Databases, GIS, and Multi-dimensional Indexing

Mapping the multi-dimensional space into the one-dimensional domain plays an important role in applications that involve multi-dimensional data. Multimedia databases, Geographic Information Systems (GIS), QoS routing, and image processing are examples of multi-dimensional applications. Modules that are commonly used in multi-dimensional applications include searching, sorting, scheduling, spatial access methods, indexing, and clustering. Numerous research has been conducted for developing efficient algorithms and data structures for these modules for one-dimensional data. In most cases, modifying the existing one-dimensional algorithms and data structures to deal with multi-dimensional data results in spaghetti-like programs handling many special cases. The cost of maintaining and developing such code degrades the system performance. Mapping from the multi-dimensional space into the one-dimensional domain provides a pre-processing step for multi-dimensional applications. The pre-processing step takes the multi-dimensional data as input and outputs the same set of data represented in the one-dimensional domain. The idea is to keep the existing algorithms and data structures independent of the dimensionality of data. The objective of the mapping is to represent a point from the D-dimensional space by a single integer value that reflects the various dimensions of the original space. Such mapping is called locality-preserving mapping in the sense that, if two points are near to each other in the D-dimensional space, then they will be near to each other in the one-dimensional space.

Network-Attached Storage Devices NASDs

Writing efficient schedulers is becoming a very challenging task given the increasing demands of such systems. Consider the case of network-attached storage devices (NASDs) [8] as a building block for a multimedia server (e. g., see [1]). NASDs are smart disks that are directly attached to the network. In a multimedia server, a major part of a NASD function goes towards fulfilling the real-time requests of users. This involves disk and network scheduling with real-time constraints, possibly with additional requirements like request priorities, and quality-of-service guarantees. NASDs requirements can be mapped in the multi-dimensional space and a SFC-based scheduler is used. The type of space-filling curve used in NASD scheduling is determined by its requirements. For example, in NASD, if reducing the number of requests that lose their deadlines is more important than increasing the disk or network bandwidth, then the real-time deadline dimension of the scheduling space will be favored. As a result, a space-filling curve with an intentional bias is favored.

Multimedia Disk Scheduling

Consider the problem of disk scheduling in multimedia servers [18]. In addition to maximizing the bandwidth of the disk, the scheduler has to take into consideration the real-time constraints of the page requests, e. g., as in the case of video streaming. If clients are prioritized based on quality-of-service guarantees, then the disk scheduler might as well consider the priority of the requests in its disk queue. Writing a disk scheduler that handles real-time and QoS constraints in addition to maximizing the disk bandwidth is challenging a task [2]. Scheduler parameters can be mapped to space dimensions and an SFC-based scheduler is used. The reader is referred to [18] to gain more insight regarding the applicability of irregularity in multi-media disk schedulers.

Future Directions

Future directions for space-filling curves include: (1) exploiting new multi-dimensional applications that can make use of the properties of space-filling curves, (2) analyzing the behavior of various space-filling curves in high-dimensional spaces, (3) providing automated modules with the ability of choosing the appropriate space-filling curve for a given application, and (4) developing new space-filling curves that are tailored to specific applications.

Recommended Reading

1. Akinlar, C., Aref, W.G., Kamel, I., Mukherjee, S.: Automatic disks: The building block for a scalable distributed file system. In: Proc. of the 5th Intl. Workshop on Multimedia Information Systems, MIS, Palm Springs Desert, October 1999
2. Aref, W.G., Kamel, I., Ghandeharizadeh, S.: Disk scheduling in video editing systems. IEEE Trans. Knowl. Data Eng., TKDE **13**(6),933–950 (2001)
3. Asano, T., Ranjan, D., Roos, T., Welzl, E., Widmayer, P.: Space-filling curves and their use in the design of geometric data structures. Theor. Comput. Sci., TCS **181**(1), 3–15 (1997)
4. Bohm, C., Klump, G., Kriegel, H.-P.: xz-ordering: A space-filling curve for objects with spatial extension. In: Proc. of 6th Intl. Symp. on Large Spatial Databases, SSD, Hong Kong, July 1999, pp. 75–90
5. Breinholt, G., Schierz, C.: Algorithm 781: Generating hilbert's space-filling curve by recursion. ACM Trans. Math. Softw., TOMS **24**(2),184–189 (1998)
6. Faloutsos, C.: Gray codes for partial match and range queries. IEEE Trans. Soft. Eng., TSE **14**(10), 1381–1393 (1988)
7. Faloutsos, C., Rong, Y.: Dot: A spatial access method using fractals. In: Proc. of Intl. Conf. on Data Engineering, ICDE, Kobe, Japan, April 1991, pp. 152–159
8. Gibson, G., Nagle, D., Amiri, K., Butler, J., Chang, F.W., Gobioff, H., Hardin, C., Riedel, E., Rochberg, D., Zelenka, J.: File server scaling with network-attached secure disks. In: Proc. of Intl. Conf. on Measurement and Modeling of Computer Systems, SIGMETRICS, Seatle, Washington, June 1997, pp. 272–284
9. Goldschlager, L.M.: Short algorithms for space-filling curves. Software–Practice and Experience, SPE **11**(1), 99–100 (1981)
10. Hilbert, D.: Über die stetige Abbildung einer Linie auf ein Flächenstück. Mathematische Annalen, 459–460 (1891)
11. Jagadish, H.V.: Linear clustering of objects with multiple attributes. In: Proc. of the intl. conf. on Management of data, SIGMOD, Atlantic City, NJ, June 1990, pp. 332–342
12. Kamel, I., Faloutsos, C.: Hilbert r-tree: An improved r-tree using fractals. In: Proc. of the 20th Intl. Conf. on Very Large Data Bases, VLDB, Santiago, Chile, September 1994, pp. 500–509
13. Lawder, J.K., King, P.J.H.: Querying multi-dimensional data indexed using the hilbert space filling curve. SIGMOD Record **30**(1), (2001)
14. Liao, S., Lopez, M.A., Leutenegger, S.T.: High dimensional similarity search with space-filling curves. In: Proc. of Intl. Conf. on Data Engineering, ICDE, Heidelberg, Germany, April 2001, pp. 615–622
15. Mandelbrot, B.B.: Fractal Geometry of Nature. W.H. Freeman, New York (1977)
16. Mellor-Crummey, J.M., Whalley, D.B., Kennedy, K.: Improving memory hierarchy performance for irregular applications. In: Proc. of the Intl. Conf. on Supercomputing, ICS, Rhodes, Greece, June 1999, pp. 425–433
17. Mokbel, M.F., Aref, W.G.: Irregularity in multi-dimensional space-filling curves with applications in multimedia databases. In: Proc. of the 2nd Intl. Conf. on Information and knowledge Management, CIKM, Atlanta, GA, November 2001, pp. 512–519
18. Mokbel, M.F., Aref, W.G., El-Bassyouni, K., Kamel, I.: Scalable Multimedia Disk Scheduling. In: Proceedings of the International Conference on Data Engineering, ICDE, Boston, MA, March 2004, pp. 498–509
19. Mokbel, M.F., Aref, W.G., Kamel, I.: Performance of Multi-Dimensional Space-Filling Curves. In: Proceedings of the ACM Symposium on Advances in Geographic Information Systems, ACM GIS, McLean, VA, November 2002. pp. 149–154
20. Mokbel, M.F., Aref, W.G., Kamel, I.: Analysis of Multi-dimensional Space-Filling Curves. GeoInformatica **7**(3), 179–209 (2003)
21. Moon, B., Jagadish, H.V., Faloutsos, C., Salz, J.: Analysis of the clustering properties of hilbert space-filling curve. IEEE Trans. Knowl. Data Eng., TKDE **13**(1), 124–141 (2001)
22. Moore, E.H.: On certain crinkly curves. Trans. Am. Math. Soc. 72–90 (1900)
23. Niedermeier, R., Reinhardt, K., Sanders, P.: Towards optimal locality in mesh-indexing. In: Proc. of the 11th Intl. Symp. on Fundamentals of Computation Theory, FCT, Krakow, Poland, September 1997, pp. 364–375
24. Orenstein, J.A.: Spatial query processing in an object-oriented database system. In: Proc. of the intl. conf. on Management of data, SIGMOD, Washington D.C., May 1986, pp. 326–336
25. Ou, C.-W., Gunwani, M., Ranka, S.: Architecture-independent locality-improving transformations of computational graphs embedded in k-dimensions. In: Proc. of the 9th ACM Intl. Conf. on Supercomputing, ICS, Barcelona, Spain, July 1995, pp. 289–298
26. Peano, G.: Sur une courbe qui remplit toute une air plaine. Mathematishe Annalen **36**, 157–160 (1890)
27. Sagan, H.: Space Filling Curves. Springer, Berlin (1994)
28. Sierpinski, W.: Sur une nouvelle courbe qui remplit toute une aire plaine. Bull. Acad. Sci. Cracovie, SerieA 462–478 (1912)
29. Witten, I.H., Wyvill, B.: On the generation and use of space-filling curves. Software – Practice and Experience **3**, 519–525 (1983)
30. Zhang, Y., Webber, R.E.: Space diffusion: An improved parallel halftoning technique using space-filling curves. In: Computer Graphics Proc., August 1993, pp. 305–312

Space-Time Interaction

► CrimeStat: A Spatial Statistical Program for the Analysis of Crime Incidents

SPAN

► Temporal GIS and Applications

Spatial

► Uncertainty, Modeling with Spatial and Temporal

Spatial Access Method

► R*-tree

Spatial Accuracy Assessment

► Uncertain Environmental Variables in GIS

Spatial Aggregate Computation

► Aggregation Query, Spatial

Spatial Analysis

► Data Analysis, Spatial

Spatial Analysis of Crime

► Crime Mapping and Analysis

Spatial and Geographically Weighted Regression

H. SEBNEM DÜZGÜN, SERKAN KEMEÇ
Geodetic and Geographic Information Technologies,
Middle East Technical University, Ankara, Turkey

Synonyms

Spatial prediction; Regression; Global and local spatial modeling; Simultaneous autoregression; Moving average regression; Conditional spatial regression

Definition

Spatial regression (SR) is a global spatial modeling technique in which spatial autocorrelation among the regression parameters are taken into account. SR is usually performed for spatial data obtained from spatial zones or areas. The basic aim in SR modeling is to establish the relationship between a dependent variable measured over a spatial zone and other attributes of the spatial zone, for a given study area, where the spatial zones are the subset of the study area. While SR is known to be a modeling method in spatial data analysis literature [1,2,3,4,5,6], in spatial data-mining literature it is considered to be a classification technique [7].

Geographically weighted regression (GWR) is a powerful exploratory method in spatial data analysis. It serves for detecting local variations in spatial behavior and understanding local details, which may be masked by global regression models. Unlike SR, where regression coefficients for each independent variable and the intercept are obtained for the whole study region, in GWR, regression coefficients are computed for every spatial zone. Therefore, the regression coefficients can be mapped and the appropriateness of stationarity assumption in the conventional regression analyses can be checked.

Historical Background

Regression analysis is one of the basic methods for modeling variation in a dependent (response, endogenous) variable (Y) based on other (covariates or independent or explanatory or predictor or exogenous) variables (X_1, X_2, ..., X_n). Owing to the nature of this technique it is mostly used in analysis of spatial data represented by spatial zones and areas, where several attributes are specified for each spatial zone or area. These spatial zones can be in regular lattice form (e. g., pixels of remotely sensed images) or irregular areas (e. g., administrative districts). As the spatial data contain autocorrelation, the lack of ability to include this property in non-SR led analysts to develop SR models for better treatment of spatial data. In this way, the elimination of the main shortcomings of non-SR, which are assumptions of identically and independently distributed (i.i.d.) explanatory variables (X_i's) and uncorrelated error terms, is attempted by relaxing the regression method with the allowance of spatial autocorrelation. While initially, SR methods were widely used in econometrics [6,9], their utilization became popular in broad scientific disciplines, such as analysis of crime, modeling land prices, poverty mapping, epidemiology, air pollution and health, natural and environmental sciences, etc.

Although SR models take the spatial variability of the parameters into account, they are not considered as local models since they are developed for modeling the mean of the spatial phenomena. For this reason, SR models have limitations for explaining the local characteristics of the spatial phenomena, especially when the phenomena vary over the space [10]. GWR is a relatively simple and effective technique for exploring spatial nonstationarity, which is characterized by changes in relationships across the study region leading to varying relations between dependent and independent variables. Hence there is a need for better understanding of the spatial processes has emerged local modeling techniques. GWR, introduced in [11], is one of the most widely used local modeling methods. Since the introduction of GWR, it has been implemented in various disciplines such as the natural, environmental, social and earth sciences.

Scientific Fundamentals

Nonspatial Regression

For p independent variables ($\boldsymbol{X}$) and dependent variables ($\boldsymbol{Y}$) with n number of spatial zone (areas) the linear regression model is:

$$\boldsymbol{Y} = \boldsymbol{X}\beta + \boldsymbol{\varepsilon} \tag{1}$$

where $\boldsymbol{Y}$ is the vector of dependent variable, $(n \times 1)$, $\boldsymbol{X}$ is the matrix of the values of p independent variables in each spatial zone, $(n \times p)$, $\boldsymbol{\varepsilon}$ is the vector of errors with zero mean and constant variance, $(n \times 1)$, and β is the regression coefficient to be estimated $(p \times 1)$.
Then, estimates of $\beta(\hat{\beta})$ and its variance by least squares approximation are given by:

$$\hat{\beta} = \left(X^{\mathrm{T}}X\right)^{-1} X^{\mathrm{T}}y \tag{2}$$

$$VAR\left(\hat{\beta}\right) = \sigma^2 \left(X^{\mathrm{T}}X\right)^{-1} . \tag{3}$$

σ^2 is usually unknown and predicted by using residuals $(\hat{\sigma}^2)$:

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n-p} \tag{4}$$

where:

$$\hat{y}_i = x_i^{\mathrm{T}} \hat{\beta} . \tag{5}$$

Estimation of residuals is used to assess the fit of the model. The overall goodness of fit is provided by coefficient of determination (R^2):

$$R^2 = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y}_i)^2} \tag{6}$$

Spatial Regression

SR is modification of the regression equation (1) by using a contiguity matrix (proximity matrix or geographic weights matrix) in which the neighborhood information about the spatial zones are characterized and accounted for spatial autocorrelation. There are basically three SR models depending on the formulation of spatial interaction: Simultaneous autoregression (SAR), moving average (MA) and conditional SR (CSR)
SAR model (7) is also called the autocorrelated error model as the error term in the non-SR model (1) is formulated in such a way that it involves spatial autocorrelation

$$\boldsymbol{Y} = \boldsymbol{X\beta} + U \tag{7}$$

where

$$U = \rho \boldsymbol{W} y + \boldsymbol{\varepsilon} . \tag{8}$$

Then:

$$\boldsymbol{Y} = \boldsymbol{X}\beta + \rho \boldsymbol{W} y + \boldsymbol{\varepsilon} \tag{9}$$

where $\boldsymbol{\varepsilon}$ is the vector of errors with zero mean and constant variance σ^2, $\boldsymbol{W}$ the proximity matrix, ρ: the interaction parameter or spatial autoregressive coefficient, and β the parameter to be estimated due to relationship between the variables.
As can be seen in (9), the SAR model requires prediction of the additional parameter ρ, which accounts for the direction and magnitude of the neighborhood effect. Hence, the SAR model is computationally costly as compared to non-SR, as it requires estimation of β and ρ (for more information refers to [1,5]).
For individual observation (y_i), (9) takes the following form

$$y_i = \rho \left(\sum_j w_{ij} y_j \right) + \varepsilon_i . \tag{10}$$

Instead of formulating the SAR model with autoregressive dependent variable ($U = \rho \boldsymbol{W} y + \boldsymbol{\varepsilon}$), if the error term in (7) is modeled as autoregressive, an MA model is obtained: ($U = \rho \boldsymbol{W\varepsilon} + \boldsymbol{\varepsilon}$). The basic difference between the two models is that while in a SAR model all the error terms are considered to be correlated in such a way that the correlation level decreases with increasing distance between the terms, in a MA model only the closest neighboring errors, which are defined by proximity matrix, $\boldsymbol{W}$, are correlated [5].
An individual observation (y_i) in MA model is:

$$y_i = \rho \left(\sum_j w_{ij} \varepsilon_j \right) + \varepsilon_i . \tag{11}$$

As can be observed from the formulation of SR models, the choice of $\boldsymbol{W}$, which is not a clear-cut issue and is usually made in an ad hoc manner, plays a critical role in the results [1,2,12]. Usually, $\boldsymbol{W}$ has binary or standardized row entries. Examples of $\boldsymbol{W}$ matrix construction schemes are: length of common boundary, intercentroid distance functions, sharing an edge, etc. The definition of $\boldsymbol{W}$ entries can be specified based on the hypotheses to be tested [2]. The most commonly implemented method is to use several alternative $\boldsymbol{W}$ matrices in regression analysis and to test the sensitivity of the results. Appropriate $\boldsymbol{W}$ matrix selection minimizes the standard error of the mean response variable estimate [12].

Geographically Weighted Regression

GWR arises from the extension of non-SR in (1) in order to obtain local parameter estimates instead of global estimates. The basic mechanism of GWR relies on obtaining separate regression equations for each spatial zone, in which a kernel centered on the area is adapted in such a way that neighboring areas are weighted based on a distance decay function [10]. In GWR, (1) is modified as:

$$Y = (\beta \otimes X)\,\mathbf{1} + \varepsilon \qquad (12)$$

where $\otimes$ is a multiplication operator serving for multiplication of each element of β with the corresponding element of X and $\mathbf{1}$ is vector of 1s [10]. For n number of spatial zones a with k number of independent variables, β and $\boldsymbol{X}$ have $n \times (k+1)$ dimensions and hence 1 has the dimension of $(k+1) \times 1$.
Then least square estimates of $\beta_i(\hat{\beta} i)$ and their variances are:

$$\hat{\beta}_i = \left(X^{\mathrm{T}} W_i X\right)^{-1} X^{\mathrm{T}} W_i Y \qquad (13)$$

$$VAR(\hat{\beta}_i) = (X^{\mathrm{T}} W_i^{-1} X)^{-1} \qquad (14)$$

where $\boldsymbol{W}_i$ is n by n weighting matrix whose off-diagonal elements are zero and diagonal elements are the geographical weighting (15).

$$W_i = \begin{bmatrix} w_{i1} & 0 & \dots & 0 \\ 0 & w_{i2} & \dots & 0 \\ \vdots & \vdots & \vdots & \\ 0 & 0 & \dots & w_{i3} \end{bmatrix} . \qquad (15)$$

The choice of $\boldsymbol{W}_i$ depends on the selection of kernel function, which may be in the form of fixed (i. e., fixed bandwidth) or adaptive kernels (i. e., varying bandwidths). A typical kernel for $\boldsymbol{W}_i$ is given in (16). For more information about kernel functions refer to [10]. Where, b is the bandwidth and d_{ij} is distance between centers of spatial zones i and j.

$$w_{ij} = \exp\left[-\frac{d_{ij}^2}{2b^2}\right] . \qquad (16)$$

Key Applications

As indicated in the previous section, SR and GWR are widely used in broad disciplines. While SR is mainly used for modeling and then predicting spatial behavior of phenomena, GWR is used for exploring the spatial heterogeneity in a spatial phenomenon.

Applications of SR

The SR model is a popular spatial data analysis technique which has been used in many application areas with space related datasets. These application areas cover many widespread disciplines and some of the most common applications are:

Housing/Real Estate Several researchers have successfully applied SR models to real property. Most of the empirical studies have focused on spatial autocorrelation in the housing market because of the availability of housing data and the increasing demands for accurate market value estimates.

Public Health/Medical Issues SR models mostly used for analyzing relation between the causes of certain public health problems such as health concerns related to environmental causes and cancer. Several environmental factors, which have spatial correlation, are related to diseases by SR models.

Public Safety/Crime Analysis SR modeling is one of the most widely used methods in public safety issues like spatial analysis of crime. As standard regression does not consider spatial autocorrelation among spatial units, SR models give better insight into the understanding of crime rates.

Economics Poverty mapping has become an increasingly important tool to improve living standards in developing countries. Classic econometric models do not take spatial correlation structure of poverty into account. Therefore, SR models are applied in order to model the distribution of poverty across regions.

S

Transportation SR models are also used in transportation based studies. Examples are mainly modeling of utilization of public transportation in relation with trip attributes, individual attributes and household attributes.

Natural and Environmental Sciences SR modeling has widespread usage in natural and environmental sciences such as ecology, environment, agriculture, hydrology, etc. An example would be analysis of ground water vulnerability to nitrate leaching from agricultural management.

Applications of GWR

GWR has been used in various disciplines such as economics, natural, environmental, social and earth sciences, where there is a need for better understanding of local behavior of parameter relations. Some of the most common applications are as follows:

Regional Development GWR is very important and beneficial method for regional development studies as a local spatial modeling technique. It could be used to find regional development disparities and nonstationarities to analyze local characteristics. Regional development analysis applications could cover all aspects of development or aim at specifically one sector like industry.

Environmental Criminology Environmental criminology deals with the interaction between crime and the space, especially spatial distribution of crime and its causes. GWR is used for examining spatial relations between residential burglaries and socioeconomic characteristics of a region. The results of GWR helps investigating local variation of the burglary risk.

Economics GWR is also used for exploring residential property valuation to explain the variation in the house prices according to variation in the independent variables. Independent variables could be age of the structure, accessibility to urban services and proximity to schools.

Political Geography The geographical analysis of elections involves explaining geographical variations in the votes and related spatial variables. The spatial variability in voting patterns and related parameters can be investigated by GWR. In this way, local heterogeneity in the voting pattern can be explored.

Agricultural Planning When conventional regression methods give coarse results in explaining the impact of a famine, GWR is used to analyze local details. Agricultural political decisions could be inappropriate due to these coarse results. Hence. detailed analysis of the heterogeneities by GWR model plays an important role for agricultural issues. Analysis of the farm locations with a spatial approach could produce useful results to improve agricultural production at a subregional scale.

Meteorological Applications Analysis of rainfall data is one of the key concepts in meteorological applications. Rainfall analysis constitutes the basis of ecology and hydrology studies. GWR estimates, in conjunction with a digital elevation matrix of the study area, have the potential to facilitate the rapid production of rainfall maps to be used by ecologists and hydrologists.

Forest Ecosystems The determination of the economic value of a forest ecosystem is a critical issue in forest ecosystems. GWR is used for exploring spatially correlated characteristics of a forest ecosystem at landscape, regional and continental scales. In this way it is possible to predict net primary production of a forest ecosystem.

Environmental Issues Land degradation, which is the process of gradual degradation or loss of biological and economical productivity of a land, is one of the key issues in environmental management. GWR is effectively used for evaluation of land performance and identification of degrading areas.

Future Directions

Although SR and GWR techniques were initially used in the social sciences, especially in econometrics and quantitative geography, their use has rapidly spread to other scientific disciplines where there is a need for spatial prediction models for spatial data collected in the form of areas or zones. This trend will continue in future, especially for the environmental and earth sciences, where remotely sensed spatial data are increasingly utilized.

Cross References

► Autocorrelation, Spatial
► Public Health and Spatial Modeling

Recommended Reading

1. Bailey, T., Gatrell, A.C.: Interactive Spatial Data Analysis. Pearson, Essex, England (1995)
2. Haining, R.: Spatial Data Analysis in the Social and Environmental Sciences. Cambridge University Press, Cambridge, UK (1993)
3. Schabengerger, O., Gotway, C.A.: Statistical Methods for Spatial Data Analysis. Chapman & Hall/CRC, Boca Raton, Florida, USA (2005)
4. Haining, R.P., Spatial Data Analysis Theory and Practice, Cambridge University Press, Cambridge, UK (2003)
5. Fotheringham, A.S., Brunsdon, C., Charlton, M.: Quantitative Geography Perspectives on Spatial Data Analysis. Sage, London, UK (2000)
6. Anselin, L.: Spatial Econometrics: Methods and Models. Kluwer, Dordrecht (1988)
7. Shekhar, S., Chawla, S.: Spatial Databases: a Tour. Pearson, New Jersey, USA (2003)
8. Cliff, A.D., Ord, J.K.: Spatial Autocorrelation. Pion, London (1973)
9. Gamerman, D., Moreira, A.R.B.: Multivariate spatial regression models. J. Multivariate Anal. **91**, 262–281 (2004)
10. Fotheringham, A.S., Brunsdon, C., Charlton, M.: Geographically Weigthed Regression the Analysis of Spatially Varying Relationships. Wiley, West Sussex, UK (2002)
11. Brunsdon, C., Fotheringham, A.S., Charlton, M.: Geographically weighted regression: a method for exploring spatial nonstationarity. Geogr. Anal. **28**, 281–98 (1996)
12. Griffith, D.A.: Some guidelines for specifying the geographic weights matrix contained in spatial statistical models. In: Arlinghous, S.L., Griffith, A.D., Drake, W.D., Nystuen, J.D. (eds.)

Practical Handbook of Spatial Statistics, pp. 65–82, CRC, Boca Raton, Florida, USA (1995)

Spatial and Spatio-temporal Queries Over Moving Objects

► Spatio-temporal Queries on Road Networks, Coding Based Methods

Spatial Anomaly Detection

► Outlier Detection, Spatial

Spatial Aspects of Crime

► Crime Mapping and Analysis

Spatial Association

► Co-location Pattern Discovery
► Patterns, Complex

Spatial Association Analysis

► Co-location Pattern Discovery

Spatial Association Pattern

► Co-location Pattern

Spatial Autocorrelation Measures

SUMEETA SRINIVASAN
School of Engineering and Applied Sciences,
Harvard University, Cambridge, MA, USA

Definition

A statistic that assesses the global clustering of spatial data.

Main Text

Moran's I and Geary's C are indices of spatial autocorrelation. A spatial contiguity matrix W_{ij}, with a zero diagonal, and the off-diagonal non-zero elements indicating contiguity of locations ***i*** and ***j*** are used to code proximities. The most commonly used of global indicators of spatial autocorrelation are Moran's I and Geary's C which are defined as:

$$I = \frac{N \sum_i \sum_j W_{ij} Z_i Z_j}{\sum_i \sum_i W_{ij} \sum_i Z_i^2}\,, \tag{1}$$

$$C = \frac{(N-1) \sum_i \sum_j W_{ij} (x_i - x_j)^2}{2 \left(\sum_i \sum_j W_{ij} \right) \sum_i Z_i^2}\,. \tag{2}$$

Z_i is the deviation of the variable of interest x_i from the mean $\bar{x}$ at location i, and N is the number of data points.

Recommended Reading

1. Cliff, A., Ord, J.K.: Autocorrelation, Spatial. Pion, London (1973)

Spatial Autoregressive Models

► Spatial Regression Models

Spatial Causality

► Dynamic Travel Time Maps

Spatial Cone Tree

► Correlation Queries in Spatial Time Series Data

Spatial Contiguity Matrices

SUMEETA SRINIVASAN
School of Engineering and Applied Sciences,
Harvard University, Cambridge, MA, USA

Synonyms

Spatial weight matrices

Definition

These are spatial matrices that are used to define spatial contiguity.

Main Text

The contiguity relations used for spatial autocorrelation measures such as Moran and Geary are in terms of sets of neighbors of location i. These are coded in the form of a weights matrix $\mathbf{W}$, with a zero diagonal, and the off-diagonal non-zero elements. The off-diagonal elements are often scaled to sum to unity in each row. They indicate the proximity of locations either through distance measures or by equaling 1 when the locations are proximate and 0 otherwise.

Cross References

► Autocorrelation, Spatial
► Spatial Autocorrelation Measures

Recommended Reading

1. Cliff, A., Ord, J.K.: Autocorrelation, Spatial. Pion, London (1973)

Spatial Correlation

► Autocorrelation, Spatial

Spatial Data, Indexing Techniques

REYNOLD CHENG
Department of Computing, Hong Kong
Polytechnic University, Hong Kong, China

Synonyms

Probability treshold indexing; Spatial data indexing with uncertainty

Definition

The *Probabilistic Threshold Indexing* (PTI) method enables efficient query evaluation of a vast amount of spatial data that are inherently uncertain. Spatial data uncertainty is common in Location-Based Services (LBS), where the positions of users are acquired and queried for providing services. These location data are often imprecise due to measurement error, sampling error, or message delay. A user may also want to protect his/her privacy by providing a less precise location. These sources of data uncertainty have to be considered in order to avoid drawing erroneous conclusions about the data being queried.

Recently, a *probabilistic query* has been proposed to evaluate data uncertainty. This query returns probabilistic guarantees that specify the likeliness that an object satisfies the query. The PTI supports a variant of probabilistic queries, called *probability threshold queries*, which return answers whose probabilities are higher than a user-defined value. The main idea of the PTI is to precompute information for each uncertain object, and augment this information in the nodes of an R-tree. The PTI provides early filtering opportunities and reduces the need to perform expensive integration operations that are required for computing probabilities.

The PTI supports various types of probability threshold queries. This chapter studies the use of PTI to answer two types of queries: the *Probabilistic Range Query* (PRQ) and the *Imprecise Location-Dependent Range Query* (IRQ). The PRQ consists of a user-defined range, where all objects that have a chance inside the range are returned. The IRQ is a kind of location-dependent query, where the user-defined range is a function of the query issuer's location. The performance of evaluating both queries can be improved significantly with the use of PTI.

Historical Background

In recent years, positioning technologies like the Global Positioning Systems (GPS), GSM, RF-ID and the Wireless LAN have undergone rapid development [27]. These technologies allow locations of users to be determined accurately, and enable a new class of applications known as *Location-Based Services* (LBS). An important LBS is the E-911 system mandated by the U.S. (correspondingly E-112 in Europe), which requires cell phone companies to provide an accurate location (i. e., within a few hundred feet) of a cell phone user who calls for emergency help [27]. Other LBS applications include downloading driving directions to a gas station, receiving an alarm when a military adversary has crossed the border, retrieving the current locations of family members, and displaying the user's location on a map. All these applications require extensive use of location data [18].

An important issue concerning the LBS is the *uncertainty* of location data. In particular, location data obtained from physical devices are inherently imprecise due to measurement error, sampling error and network latency [6,11,23]. Some recent works (e. g., [2,10,13]) have suggested that location privacy can be protected by injecting a controlled degree of spatial uncertainty into location data, so that the actual location is hidden. In many situations, therefore, it is often impossible for the query processing server to get an accurate location value. It is thus reasonable to use a location uncertainty model to describe the imprecision of the data values, and evaluate the location uncertainty in order to provide probabilistic guarantees over the validity of the query answers. A common model for characterizing location uncertainty of an object is a closed region together with a probability distribution function (pdf) of the object in the region [11,23] or the road network [1]. These types of data are evaluated by *probabilistic queries* in order to compute the probability that each location object satisfies a query.

A number of common probabilistic queries designed for LBS have been recently investigated. For example, probabilistic range queries, which return probabilities that objects are located within a specific range, have been studied in [11,23]. In [6], the computation of probabilities that each location object satisfies a nearest-neighbor query is presented. Probabilistic join algorithms over two uncertain data sets are studied in [8]. The evaluation and quality of different probabilistic queries in an uncertain database is studied in [5].

Since probabilistic queries may involve numerical integration operations which can be costly, a number of indexing methods have been developed. These methods use some spatial data structures (e. g., R-tree [14]) in order to perform low-cost pruning over a large amount of uncertain objects. As a result, only a small fraction of objects need to have their probabilities computed (or refined). In particular, efficient indexing methods for probabilistic range queries in one-dimensional and multi-dimensional space are studied in [9] and [25]. Indexing methods for probabilistic nearest queries are studied in [6], and those for join queries are presented in [8]. Many of these methods exploit the fact that users are only interested in answers whose probabilities are higher than some user-defined threshold. This chapter explains how probabilistic range queries (PRQ), with probability threshold constraints, can be evaluated efficiently.

In LBS, a kind of well-studied query is the *location-dependent query*, which considers the location of the user who issues the query (called "query issuer") in order to decide the query result [12,16,18]. For example, the user may want to know the available cabs within two miles of his/her current location. In addition to the uncertainty of the data being queried, the imprecision of the query issuer's location further affects the validity of the query answer. A number of recent works investigate this issue. For example, Song et al. [24] study the evaluation of a a continuous nearest-neighbor query for retrieving the nearest neighbors for all points on a line segment. Their algorithm is further improved in [26]. In [15], the range nearest-neighbor query is proposed, which retrieves the nearest neighbor for every point in a multi-dimensional rectangular range. In these works, although the query issuer's location is imprecise, the data being queried has no uncertainty. Recently, [21] considers both query and data uncertainty for nearest-neighbor queries. The computation of probability values for location-dependent range queries is investigated in [4].

This chapter discusses an indexing solution for imprecise location-dependent range queries (IRQ), where the uncertain nature of both positions of the query issuers and the objects being queried are considered.

Scientific Fundamentals

To investigate how PTI provides scalable data retrieval for probabilistic threshold queries, the spatial uncertainty model is first explained in the sect. "Probabilistic Uncertainty Model". The basic structure of the PTI is then presented in the sect. "Probability Threshold Indexing (PTI)". The use of PTI for providing efficient solutions for PRQ and IRQ are presented in sects. "Probabilistic Range Queries (PRQ)" and "Imprecise Location-Dependent Range Queries (IRQ)", respectively.

Probabilistic Uncertainty Model

To capture location uncertainty, a data scheme known as *location uncertainty model* was proposed in [11] and [23]. This model assumes that each location data item can be represented by a region of possible values and their distributions. Formally, given an object O_i (which are called *uncertain object*), the *probabilistic uncertainty* of the two-dimensional location of $O_i (i = 1, \ldots, n)$ consists of two components:

Definition 1 The **uncertainty region** of O_i, denoted by U_i, is a closed region within which O_i is located.

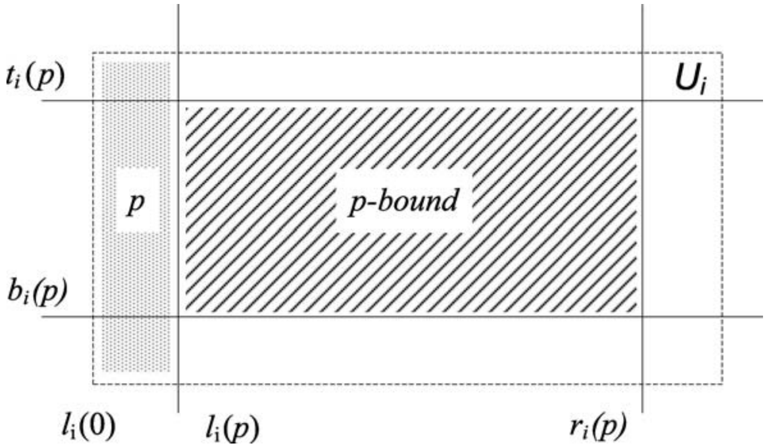
Definition 2 The **uncertainty probability density function (pdf)** of object O_i, denoted by $f_i(x, y)$, is a pdf of O_i's location, that has a value of 0 outside U_i.

Since $f_i(x, y)$ is a pdf, it has the property that $\int_{U_i} f_i(x, y)\, dx\, dy = 1$. The formula for the uncertain pdf is application-specific. Wolfson et al. propose that the object location follows the Gaussian distribution inside the uncertainty region [23]. An important type of uncertainty pdf is the uniform distribution [11], that is, $f_i(x, y) = \frac{1}{U_i}$; essentially, this implies a "worst-case" scenario where there is no knowledge of which point in the uncertainty region possesses a higher probability. The solutions presented here are applicable to any form of uncertainty pdf.

Probability Threshold Indexing (PTI)

To present the basic idea of the Probability Threshold Indexing (PTI) technique, first the concept of p-bounds – the building blocks of PTI, is explained in the sect. "The p-bound". The structure of the PTI is discussed in the sect. "Structure of PTI". The implementation issues of PTI is then presented in the sect. "PTI Implementation"

S



Spatial Data, Indexing Techniques, Figure 1 Illustrating the p-bound of O_i

The *p*-Bound A p-bound of an uncertain object O_i is a function of p, where $p \in [0, 0.5]$. It is composed of four line segments, namely $l_i(p), r_i(p), t_i(p), b_i(p)$ in 2D space, as illustrated by the hatched region in Fig. 1. The requirement of $l_i(p)$ is that the probability of the location of O_i to the left of $l_i(p)$ has to be exactly equal to p (as shown by the shaded area). Similarly, the probability of O_i to the right of the line $r_i(p)$ is exactly equal to p. The remaining line segments ($t_i(p)$ and $b_i(p)$) are defined analogously. Using the definition of p-bounds, the boundary of U_i can be represented by $l_i(0), r_i(0), t_i(0)$ and, $b_i(0)$. It will be shown that if p-bounds have been pre-computed and stored for any value of p, better pruning power can be achieved.

In practice, it is not possible to pre-compute a p-bound for each value of p. Instead, a "U-catalog" is created for each object, which is a table of a small fixed number M of tuples $\{m_k, m_k\text{-bound}\}$, where $m_k \in [0, 1]$ and $k = 1, \ldots, M$ [25]. The tuples in the U-catalog can then used for query pruning. Next, it is shown how a PTI is constructed from uncertain objects augmented with U-catalogs.

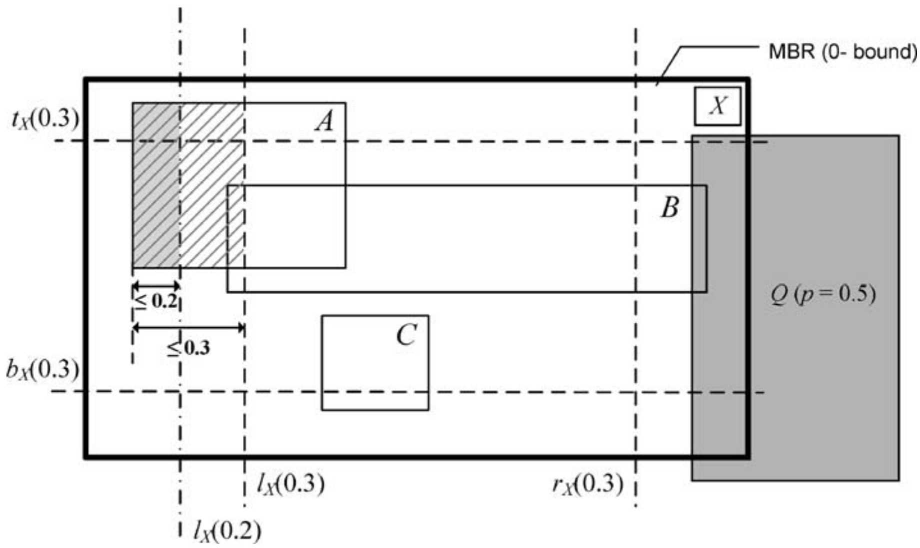
Structure of PTI The main goal of *Probability Threshold Indexing* (PTI) is to to utilize uncertainty information during an index search. It is designed based on the R-tree, which can be considered as an extension of the B-tree for indexing multi-dimensional rectangular objects [14]. The R-tree can index the uncertain regions of uncertain objects, which are rectangular in shape. Each intermediate entry of an R-tree is represented as a *minimum bounding rectangle* (MBR), which tightly bounds all the data in its subtree.

The PTI indexes uncertain objects by augmenting probability information in the R-tree's internal nodes, as shown in Fig. 3. In addition to the outermost boundary of U_i (i. e., $l_i(0), r_i(0), t_i(0), b_i(0)$), the m_k-bounds listed in the U-catalog precomputed in each object are also indexed. Specifically, in each intermediate node of the PTI, the MBR for each probability value m_k in the U-catalog is stored. Let MBR(m_k) be the minimum bounding rectangle that encloses all the m_k-bounds for all children in the node. As an example, consider a node X that consists of two objects, O_1 and O_2, where 0.3-bounds have been computed in their respective U-catalogs. The left sides of their 0.3-bounds are $l_1(0.3)$ and $l_2(0.3)$. If $l_1(0.3)$ is on the left of $l_2(0.3)$, then the left side of the 0.3-bound for node X (denoted as $l_X(0.3)$) is exactly equal to $l_1(0.3)$.

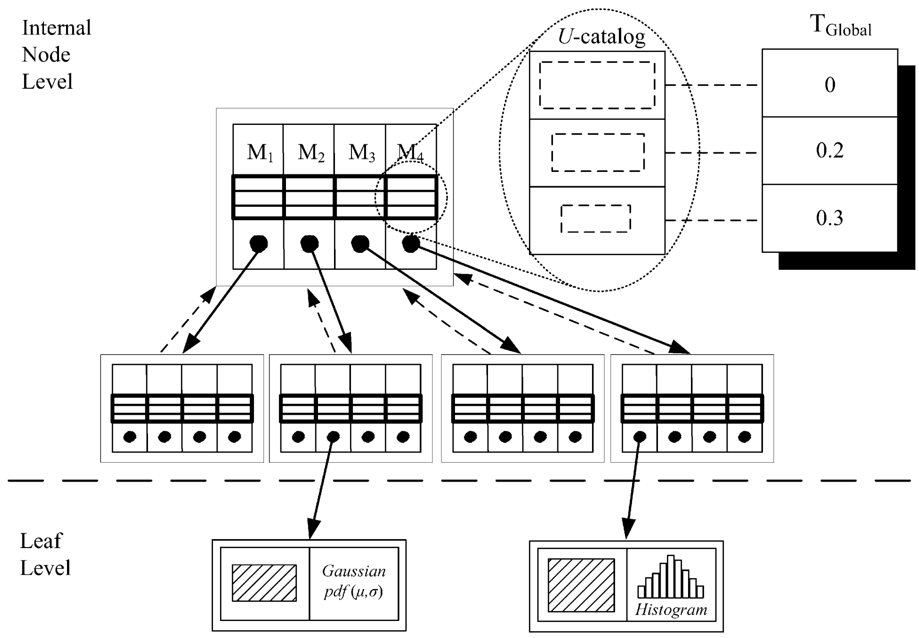
Figure 2 shows the left side of a 0.2-bound, where at most a fraction of 0.2 of each uncertain object in a sub-tree crosses it. The probability that the uncertain object A is on the left of $l_X(0.2)$ (indicated by the shaded area) is less than 0.2, while its probability of being on the left of $l_X(0.3)$ (shown by the hatched area) is less than 0.3. The uncertain object B has a chance of less than 0.3 being on the right of $r_X(0.3)$. The uncertain region of C crosses neither $l_X(0.3)$ nor $r_X(0.3)$, implying that it will have a chance of less than 0.3 of being on the left(right) of $l_X(0.3)$ ($r_X(0.3)$).

Figure 2 also illustrates that the 0-bound is exactly the MBR of node X. More importantly, for any $m_k > 0$ in the U-catalog, the m_k-bound is always enclosed by the 0-bound. In general, given two values a and b in the U-catalog, $a \geq b$ if and only if the a-bound is enclosed by the b-bound. In Fig. 2, for example, $l_X(0.3)$ is on the left of $l_X(0.2)$ since the 0.3-bound is enclosed by the 0.2-bound. This property will be revisited when the evaluation of PRQ is explained. The implementation issues of the PTI will now be examined.

PTI Implementation Figure 3 illustrates the implementation of the PTI. It has a similar design as that of the R-tree where each internal node consists of children MBRs and their corresponding pointers. In addition, each child consists of a U-catalog that stores the information of its m_k-bounds. The i-th entry of the U-catalog is a tuple containing the dimension details of the m_k-bound (as indicated by the dash-line in Fig. 3). A global table called



Spatial Data, Indexing Techniques, Figure 2 Inside an MBR of node X, with a 0.2-bound and 0.3-bound. A PRQ called Q_1 is also shown



Spatial Data, Indexing Techniques, Figure 3 Structure of PTI

T_{Global} contains the values of m_k for the m_k-bounds. In Fig. 3, for example, the 0.3-bound is the smallest rectangle of the U-catalog. The U-catalog must contain at least one entry – the 0-bound, or the MBR of the node. For convenience, it is denoted that $m_0 = 0$, i. e., the m_0-bound is the node's MBR. The data items being indexed are essentially the uncertainty regions and uncertainty pdfs, which contain Gaussian and histogram pdfs in Fig. 3.

Insertion To insert an uncertain object O_i, first compute all its m_k-bounds corresponding to the values of m_k in T_{Global}. Then, insert O_i to the PTI using a similar procedure as inserting a rectangular region to an R-tree. The main difference is that the m_k-bounds of the intermediate nodes being traversed have to be expanded appropriately in order to contain the m_k-bound of the newly inserted object. Finally, the m_k-bound information computed for O_i is copied to the U-catalog of the node that directly points to O_i, as shown in Fig. 3.

Deletion Removing an uncertain object follows a similar procedure of the R-tree. Again, it is necessary to update the m_k-bounds. In particular, if an MBR Y is to be deleted, then all the m_k bounds of the parent node that point to Y have to be shrunk to contain the m_k-bound of all MBRs in the same node as Y. Therefore, it is necessary

S

to keep parent pointers in each node (as illustrated by the dashed arrows in Fig. 3). To update the m_k-bounds, beginning from the leaf node that contains the object of interest, the changes to m_k-bounds are propagated until the root is reached.
Although the fan-out of a PTI node is lower than an R-tree node because each node contains fewer spaces to store MBRs (assume the node size is fixed), the fan-out only logarithmically affects the height of the tree. Hence, in most cases this results in increase in height by an additive constant which only has a minor effect on the PTI's performance. Indeed, its performance illustrates significant improvements over an R-tree, as detailed in the experimental results in [9]. [25] discusses the issue of how to save the overhead of the U-catalog for multi-dimensional rectangles. The following section examines how PTI can be used to support two major types of queries: PRQ and IRQ.

Probabilistic Range Queries (PRQ)

As mentioned before, the user of the PRQ is only interested in answers with probabilities higher than some pre-defined thresholds. Formally, the PRQ is defined as follows [9,25]:

Definition 3 Given a two-dimensional closed region R, a **Probabilistic Range Query** (PRQ) returns a set of tuples $\{(O_i|i \in [1, n]\}$, such that the probability that O_i is inside R (called qualification probability p_i) is greater than or equal to the probability threshold P, where $0 < P \leq 1$.

A straightforward solution to evaluate a PRQ is to first retrieve all O_i's, whose uncertainty regions overlap with R into a set S. Each O_i in S is then evaluated for their probability of satisfying the PTQ with the following operation:

$$p_i = \int_{OI} f_i(x, y)\,\mathrm{d}x\,\mathrm{d}y \qquad (1)$$

where OI is the region of overlap between R and U_i. The answer to the PRQ only includes O_i's whose qualification probabilities (p_i's) are larger than P.
There are two problems with this approach. First, how can the elements of S, i. e., U_i's that overlap with R be found? The solution can be extremely inefficient if each item O_i is retrieved from a large database and tested against R. A typical solution is to build an index structure (e. g., R-tree) over the boundaries of U_i's and apply a range search of R over the index [17,19,20]. This section reviews how a range query is performed on an R-tree. Starting from the root node, the query interval R is compared with the maximum bounding rectangle (MBR) of each child in the node. Only children with MBRs that overlap with R are further followed. By only retrieving nodes whose MBRs overlap R, some amount of I/O and computation can be saved.
The second problem is that the probability of each element in S needs to be evaluated with Eq. 1. This can be a computationally expensive operation unless the uncertainty pdf is a simple function (e. g., uniform distribution). Notice that the bottleneck incurred in this step is independent of whether an R-tree is used or not. In fact, the R-tree does not help much if many items overlap with R, but most of them have a probability less than P. In this situation, it is not necessary to spend a lot of time computing the probability values for a vast number of items only to find that they do not satisfy the PTQ after all.

Answering PRQ with PTI The two problems just mentioned can be improved with the use of PTI. In particular, the U-catalog residing in the intermediate node can be used to avoid its contents from being further investigation. This can result in significant savings in the number of I/Os. Furthermore, it is not necessary to compute the probability values of the uncertain objects which cannot satisfy the query anyway. To illustrate how this idea works, it is necessary to refer to Fig. 2 again. Here, a range query Q, represented as a shaded rectangular area, is tested against the U-catalog of the internal node. Without the aid of the U-catalog, Q has to (i) examine which MBR(s) (i. e., A, B, or C) overlap(s) with Q's region, (ii) for the qualified MBRs (B in this example), further retrieve the node pointed by B until the leaf level is reached, and (iii) compute the probability of the interval in the leaf level.
The presence of the U-catalog makes it possible to easily determine whether an internal node contains any qualifying MBRs without further probing into the subtrees of this node. The example of Fig. 2 tests Q's range against the four lines of node X's 0.3-bound (i. e., $l_X(0.3)$, $r_X(0.3)$, $t_X(0.3)$, $b_X(0.3)$). As shown in Fig. 2, it intersects none of these bounds. In particular, although Q overlaps the MBR, its overlapping region is somewhere between $r_X(0.3)$ and the right boundary of the node X's MBR. Recall from the definition of the PTI that an uncertain object cannot have a probability of more than 0.3 for being outside the 0.3-bound. Therefore, the probability of uncertain objects in node X's MBR that fall into the range of Q cannot be larger than 0.3. Suppose that Q has a probability threshold of 0.5, i. e., Q only accepts uncertain objects with a probability of at least 0.5. Then, it is possible to be certain that *none* of the uncertain objects in the MBR satisfy Q without further probing the subtrees of this node. Compared with the case where no m_k-bounds are provided, this represents a significant saving in terms of number of I/Os and computation time. In general, given an m_k-bound of a node X, X can be elimi-

nated from further examination if the following two conditions hold:

1. R is on the outside of m_k-bound, and
2. $P \geq m_k$

If no m_k-bound in X satisfies these two conditions, the checking of intersections with X is resumed, and the range searching process is done in the same manner for an R-tree. The PTI can be used to support other queries as well. The following section examines how PTI is used to facilitate the execution of another query – the IRQ.

Imprecise Location-Dependent Range Queries (IRQ)

The second query investigated is also a variant of range query. Its main difference from PRQ is that the query range is a function of the position of the query issuer which can also be imprecise. Denote O_0 as the identity of the query issuer, and $O_1, \ldots, O_n$ as the identities of the objects being queried. Additionally, assume that the uncertainty regions of all objects are axis-parallel rectangles. For convenience, an object with no uncertainty is called a *point object*, denoted by S_i, whose location is exactly a point (x_i, y_i) in the 2D space. Then, given an axis-parallel rectangle $R(x, y)$ with center (x, y), half-width w and half-height h, and the location of O_0 as its center, the Imprecise Location-Dependent Range Query (or IRQ) is defined as follows:

Definition 4 An **Imprecise Location-Dependent Range Query** (IRQ) returns a set of tuples $\{(O_i | i \in [1, n]\}$, such that the probability p_i that O_i is inside $R(x, y)$, with $(x, y) \in U_0$, is greater than or equal to the probability threshold P, where $0 < P \leq 1$.

Figure 4 illustrates an IRQ. For convenience, a R is used to represent $R(x, y)$.

A naive solution for evaluating IRQ is to examine the probability that an uncertain object O_i satisfies the query at each point in U_0 [4]. This is given by integrating the uncertainty pdf of O_i in the overlapping area of U_i and $R(x, y)$, i. e.,

$$p_i(x, y) = \int_{U_i \cap R(x,y)} f_i(x, y)\,dx\,dy\,. \tag{2}$$

Figure 4 shows that the probability of O_1 satisfying $R(x_0, y_0)$, given that O_0 is at point (x_0, y_0), is the integral of $f_1(x, y)$ over the shaded region. Considering the uncertainty pdf of O_0, the following gives the general formula for computing p_i.

$$p_i = \int_{U_0} p_i(x, y) f_0(x, y)\,dx\,dy\,. \tag{3}$$

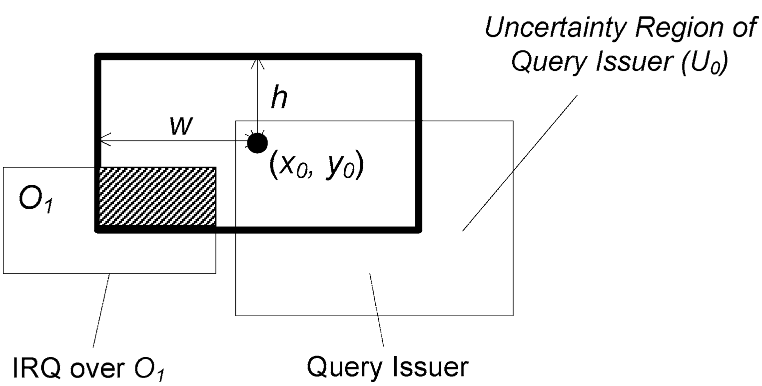
Similar to PRQ (c.f. Eq. 1), Eq. 3 is also costly to evaluate; it may necessitate the use of numerical integration. For example, in order to obtain p_i in Eq. 3, U_0 is first represented by a set of sampling points; for each sampling point, Eq. 3 is evaluated. A large number of sampling points will be needed to produce an accurate answer. This is required even if the uncertainty pdf is as simple as a uniform distribution.

Novel solutions were developed to improve this situation. The rest of this section describes how such operations can be avoided. The two techniques developed in [4] are described: (1) expanding the range query, and (2) exploiting the duality between the locations of the query issuer and data being queried. These techniques are then examined, and it is discussed how they can be applied together with PTI to answer such queries.

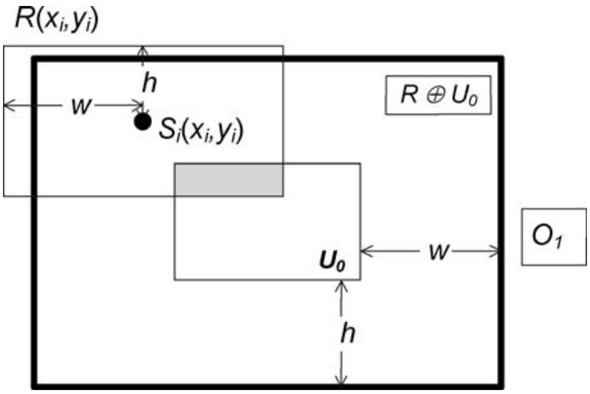
Query Expansion The first technique performs an inexpensive filtering over objects that have no chance of being included in the query answer. The main idea is to expand the query range R with the query issuer's position information. Any object that does not touch this expanded query range (called the *Minkowski Sum* [3]) can be pruned.

Lemma 1 The qualification probability of an uncertain object is non-zero if and only if its uncertainty region lies within (overlaps) the Minkowski Sum of R and U_0.

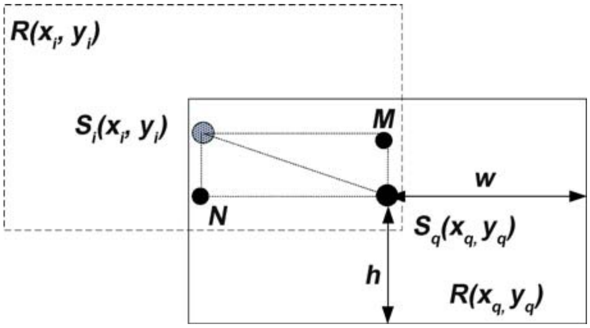
S



Spatial Data, Indexing Techniques, Figure 4
Illustrating IRQ



Spatial Data, Indexing Techniques, Figure 5 Illustrating the evaluation of IRQ. The thick line is the expanded query using the Minkowski Sum



Spatial Data, Indexing Techniques, Figure 6 The Duality of Query and Data

Before Lemma 1 can be explained, it is necessary to first review the Minkowski Sum [3]: $A \oplus B = \{x + y | x \in A, y \in B\}$ where A and B are two given polygons, and x and y are points. Conceptually, the Minkowski Sum is the union of all translations of A by a point y located in B. It is possible to view the Minkowski Sum of the query range R and the uncertainty region U_0, that is, $R \oplus U_0$, as the union of all range queries by considering all the possible positions of O_0 who reside somewhere in U_0. If the uncertainty region of any object being queried does not overlap $R \oplus U_0$, it creates assurance that this object does not have any chance of satisfying any range query issued at any position in U_0. Thus, $R \oplus U_0$ is used as a query range to obtain objects that have a non-zero qualification probability of satisfying the IRQ (i. e., their uncertainty regions overlap with the query range).

Figure 5 illustrates the Minkowski Sum of R and U_0 which can simply be obtained by extending U_0 by w on the left and right, and by h on the top and bottom.[1] Hence, the Minkowski Sum can be derived in a linear time. As shown in the same figure, the expanded query range allows objects with zero qualification probability (i. e., objects O_1) to be pruned immediately. In fact, if the objects are indexed by an R-tree, then any nodes whose MBRs do not intersect with the Minkowski Sum can be pruned immediately. The pruning performance can be further improved by utilizing the PTI. Before explaining how this is possible, it is necessary to introduce the concept of *query-data duality*.

Query-Data Duality The second method exploits the fact the role of the query issuer and the data being queried can be exchanged. Specifically, Lemma 2 describes this "query-data duality" property:

Lemma 2 (Query-Data Duality) Given two point objects S_i and S_q (with locations (x_i, y_i) and (x_q, y_q) respectively), S_i satisfies $R(x_q, y_q)$ if and only if S_q satisfies $R(x_i, y_i)$.

Proof: Construct a rectangle with vertices M, S_i, N, and S_q, as shown in Fig. 6. If S_i satisfies $R(x_q, y_q)$, then $|S_iM| \le w$ and $|S_iN| \le h$. This implies $|S_qN| \le w$ and $|S_qM| \le h$. Hence, S_q must satisfy the query $R(x_i, y_i)$. Conversely, if S_q satisfies $R(x_i, y_i)$, the same rectangle can be reconstructed and it is thus proven that $|S_iM| \le w$ and $|S_iN| \le h$. Hence, S_i must also satisfy $R(x_q, y_q)$. □

In other words, if a point object S_i satisfies the range query issued by S_q, then S_q must also satisfy the range query issued by S_i. This leads to the following result.

Lemma 3 The qualification probability p_i of a point object S_i for satisfying an IRQ can be computed by

$$\int_{R(x_i,y_i)\cap U_0} f_0(x, y)\,\mathrm{d}x\,\mathrm{d}y\,. \tag{4}$$

Proof: Consider the overlapping region of U_0 and the range query $R(x_i, y_i)$ issued by S_i, as shown in the shaded area in Fig. 5. Obviously, any point $(x_e, y_e) \in U_0 \cap R(x_i, y_i)$ satisfies the query $R(x_i, y_i)$. Using Lemma 2, S_i also satisfies the range query $R(x_e, y_e)$. Moreover, S_i does not satisfy any range queries centered at points outside the overlapping region. Hence, only the queries issued at points (x_e, y_e) can have S_i in their answer, and the qualification probability of S_i is simply the integration of the uncertainty pdf of O_0 in the overlapping region, i. e., $\int_{R(x_i,y_i)\cap U_0} f_0(x, y)\,\mathrm{d}x\,\mathrm{d}y$. □

Lemma 3 can now be used to compute the qualification probability of an uncertain object O_i for satisfying the IRQ. Specifically, treat every point $(x, y) \in U_i$ as a point object, and compute the qualification probability of

[1] If U_0 and R are m-sided and n-sided polygons, the Minkowski Sum is a convex polygon with at most $m + e$ edges, which requires $O(m + e)$ time to compute [3].

each individual point (x, y) for satisfying the IRQ (termed $Q(x, y)$), with Lemma 3. The qualification probability of O_i is then simply equal to the integral of all these $Q(x, y)$ values, weighted by the uncertainty pdf of O_i at (x, y), i. e.,

$$p_i = \int_{U_i} f_i(x, y) \cdot Q(x, y) \, \mathrm{d}x \, \mathrm{d}y \,. \tag{5}$$

Hence, Eq. 5 provides an alternative to Eq. 3 for computing IRQ. Although it is not immediately clear which method is better, note that the performance of Eq. 5 can be further improved when combined with the results about query expansion.

Lemma 4 The qualification probability p_i of an uncertain object O_i for satisfying an IRQ can be computed by

$$p_i = \int_{U_i \cap (R \oplus U_0)} f_i(x, y) \cdot Q(x, y) \, \mathrm{d}x \, \mathrm{d}y \,. \tag{6}$$

The only difference between this equation and Eq. 5 is that U_i is replaced by $U_i \cap (R \oplus U_0)$ – which potentially produces a smaller integration region and better computational performance. How is this possible? Observe that for any point $(x_t, y_t) \in U_i - (R \oplus U_0)$, $Q(x_t, y_t)$ must be zero, according to Lemma 1. Hence, it is fine to focus on the portion of U_i that overlaps the expanded query region.
It is now explained how these results can be used together with the PTI to improve the performance of IRQ.

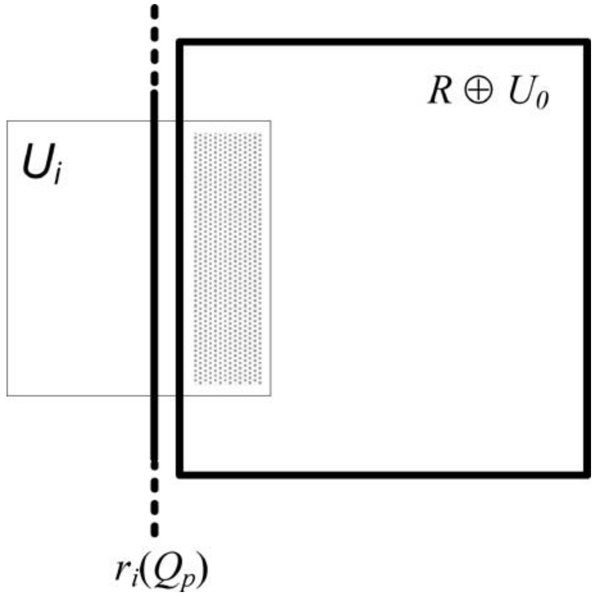
Answering IRQ with *p*-Bounds and PTI The p-bound of O_i can be used to facilitate the pruning of O_i for IRQ. Observe that O_i can be pruned if the common region of U_i and $R \oplus U_0$ is on the right of $r_i(P)$. Figure 7 assures that $\int_{U_i \cap (R \oplus U_0)} f_i(x, y) \, \mathrm{d}x \, \mathrm{d}y \le P$. From Lemma 4:

$$p_i = \int_{U_i \cap (R \oplus U_0)} f_i(x, y) \cdot Q(x, y) \, \mathrm{d}x \, \mathrm{d}y \tag{7}$$

$$\le \int_{U_i \cap (R \oplus U_0)} f_i(x, y) \, \mathrm{d}x \, \mathrm{d}y \tag{8}$$

$$\le P \,. \tag{9}$$

Therefore, O_i can be removed from the result. If $r_i(P)$ cannot be found, it is shown that the maximum value V in the U-catalog of O_i such that $V \le P$, and then use $r_i(V)$ instead of $r_i(P)$ in the pruning process. Notice that $r_i(V)$ is on the right side of $r_i(P)$, therefore, it may be possible that the shaded region crosses $r_i(V)$ but not $r_i(P)$. The same idea can be applied to other dimensions. For example, if $U_i \cap (R \oplus U_0)$ is at the lower part of $b_i(P)$, U_i can be pruned.
With the aid of the PTI, the above pruning techniques can be applied in the index level. In particular, the same techniques can be used with the U-catalog that resides in the intermediate node. This is because every p-bound stored in this U-catalog must enclose the p-bounds for each children. Moreover, the pruning techniques rely on the fact that the uncertainty region of the objects lie outside the p-bound. In Fig. 7, for example, if $R \oplus U_0$ is on the right side of the m-bound of the U-catalog in the intermediate node, $R \oplus U_0$ must also be on the right side of $r_i(Q_p)$, assuming U_i is stored under that intermediate node. In other words, if the p-bound in the intermediate node level satisfies the pruning condition, so does its children. The children under the same subtree can then be immediately pruned.
Another technique for improving the performance of IRQ, based on shrinking the size of the query range, is discussed in [4]. The details are omitted here due to space limitations.



Spatial Data, Indexing Techniques, Figure 7 Pruning O_i for IRQ using the $r_i(P)$ bound of O_i

Key Applications

The two queries described here, namely PRQ and IRQ, have a wide range of applications in location-based services. A PRQ can be used to check whether a military adversary has crossed the border. An IRQ can help a user to check the available cabs within two miles of his/her current location. It also helps a policeman to look for suspect vehicles within some distance from his location. Another application of IRQ is to protect the location privacy of the query user who is only willing to report his/her location with a coarse granularity. The IRQ provides query evaluation facilities for these users who want to hide their actual locations from the service provider.

The incorporation of uncertainty in computing qualification probabilities is important for providing guarantees over the correctness of query answers. Uncertainty is commonly found in spatial, geographical data, and, therefore, they have to be considered in order to avoid producing false results. However, evaluating these queries can be computationally expensive. The indexing techniques described in this chapter enable a scalable evaluation of uncertainty in a large spatial database, and avoid the costly calculation of qualification probabilities.

Future Directions

An future work in this area involves the study of how indexing techniques can be used to support other kinds of probabilistic queries that are commonly found in LBS. These include spatial joins, nearest-neighbor queries and reverse-neighbor queries. Another extension involves the evaluation of continuous queries – queries that reside in the system for an extensive amount of time and are commonly found in location-based services.

Recently, a number of uncertain database prototypes have been developed (e. g., [7,22,28]). It is anticipated that uncertainty management will be become a convenient facility in future spatial database systems. This involves enhancing a database system to support user-friendly and efficient management of uncertain data. Further work is required to revise query languages and evaluation techniques in order to fulfill these visions.

Cross References

► Approximation
► Imprecision and Spatial Uncertainty
► Indexing the Positions of Continuously Moving Objects
► Moving Object Uncertainty
► Positional Accuracy Improvement (PAI)
► Spatial Data, Indexing Techniques
► Spatial Uncertainty in Medical Geography: A Geostatistical Perspective
► Uncertain Environmental Variables in GIS
► Uncertainty, Semantic

Recommended Reading

1. Almeida, V., Güüting, R.: Supporting uncertainty in moving objects in network databases. In: Proc. GIS, 2005
2. Beresford, A.R., Stajano, F.: Location Privacy in Pervasive Computing. IEEE Pervasive Computing, **2**(1) (2003)
3. Berg, M., Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry – Algorithms and Applications, 2nd edn. Springer Verlag (2000)
4. Chen J., Cheng, R.: Efficient evaluation of imprecise location-dependent queries. In: Proc. IEEE ICDE, 2007
5. Cheng, R., Kalashnikov, D., Prabhakar, S.: Evaluating probabilistic queries over imprecise data. In: Proc. ACM SIGMOD, 2003
6. Cheng, R., Kalashnikov, D.V., Prabhakar, S.: Querying imprecise data in moving object environments. IEEE TKDE, **16**(9), September 2004
7. Cheng, R., Singh, S., Prabhakar, S.: U-DBMS: An uncertainty database management system. In: Proc. VLDB, 2004
8. Cheng, R., Singh, S., Prabhakar, S., Shah, R., Vitter, J., Xia, Y.: Efficient join processing over uncertain data. In: Proc. ACM CIKM, 2006
9. Cheng, R., Xia, Y., Prabhakar, S., Shah, R., Vitter, J.S.: Efficient indexing methods for probabilistic threshold queries over uncertain data. In: Proc. VLDB, 2004
10. Cheng, R., Zhang, Y., Bertino, E., Prabhakar, S.: Preserving user location privacy in mobile data management infrastructures. In: Proc. of the 6th Workshop on Privacy Enhancing Technologies, June 2006
11. Pfoser D., Jensen, C.: Capturing the uncertainty of moving-objects representations. In: Proc. SSDBM, 1999
12. Gedik B., Liu, L.: Mobieyes: Distributed processing of continuously moving queries on moving objects in a mobile system. In: EDBT, 2004
13. Gedik B., Liu, L.: A customizable k-anonymity model for protecting location privacy. In: ICDCS, 2005
14. Guttman, A.: R-trees: A dynamic index structure for spatial searching. Proc. of the ACM SIGMOD Int'l. Conf., 1984
15. Hu H., Lee, D.: Range nearest-neighbor query. IEEE TKDE, **18**(1), 2006
16. Ilarri, S., Mena, E., Illarramendi, A.: Location-dependent queries in mobile contexts: Distributed processing using mobile agents. ACM TMC, **5**(8), August 2006
17. Kriegel, H., Potke, M., Seidl, T.: Managing intervals efficiently in object-relational databases. In: VLDB, 2000
18. Lee, D., Xu, J., Zheng, B.: Data management in location-dependent information services. IEEE Pervasive Computing, **1**(3), 2002
19. Manolopoulos, Y., Theodoridis, Y., Tsotras, V.: Chapter 4: Access methods for intervals. In: Advanced Database Indexing. Kluwer, 2000
20. Matousek, J.: Geometric range searching. ACM Computing Survey, **26**(4), 1994
21. Mokbel, M.F., Chow, C.-Y., Aref, W.G.: The new casper: Query processing for location services without compromising privacy. In: VLDB, 2006
22. Singh, S., Mayfield, C., Cheng, R., Prabhakar, S.: The ORION database (http://www.orion.cs.purdue.edu)
23. Sistla, P.A., Wolfson, O., Chamberlain, S., Dao, S.: Querying the uncertain position of moving objects. In: Temporal Databases: Research and Practice. (1998)
24. Song Z., Roussopoulos, N.: K-nearest neighbor search for moving query point. In: Proc. SSTD, 2001
25. Tao, Y., Cheng, R., Xiao, X., Ngai, W.K., Kao, B., Prabhakar, S.: Indexing multi-dimensional uncertain data with arbitrary probability density functions. In: Proc. VLDB, 2005
26. Tao, Y., Papadias, D., Shen, Q.: Continuous nearest neighbor search. In: Proc. VLDB, 2002
27. Warrior, J., McHenry, E., McGee, K.: They know where you are. Spectrum, **40**(7):20–25, July 2003
28. Widom, J.: Trio: A system for integrated management of data, accuracy, and lineage. In: Proc. CIDR, 2005

Spatial Data Indexing with Uncertainty

► Spatial Data, Indexing Techniques

Spatial Data Mining

► Geographic Knowledge Discovery
► Homeland Security and Spatial Data Mining

Spatial Data Quality

► Uncertain Environmental Variables in GIS

Spatial Data Transfer Standard (SDTS)

PHYLLIS ALTHEIDE
U.S. Geological Survey Geospatial Information Office, Rolla, MO, USA

Synonyms

SDTS; FIPS PUB 173; FIPS 173; ANSI NCITS 320-1998; Standards, consentual; NIST; FGDC; Data quality; Map accuracy; G-ring; G-polygon; Chain; Layer; Raster; Manifold; Voxel; Pixel; Profiles; Profile, topological; Vector (TVP); Digital line graph; DLG

Definition

The Spatial Data Transfer Standard (SDTS) is a standard whose purpose is to promote and facilitate the data exchange of digital geo-referenced spatial data, attributes, and metadata between different computer systems and applications

The SDTS was first ratified in 1992 by the US Department of Commerce National Institute of Standards and Technology (NIST) as Federal Information Processing Standards Publication 173. SDTS was then promoted to become a national standard and it was subsequently ratified by the American National Standards Institute (ANSI) in 1998 as National Committee for Information Technology Standards (NCITS) 320-1998. SDTS is also endorsed by the Federal Geographic Data Committee (FGDC) as one of the core standards to support the National Spatial Data Infrastructure (NSDI).

Historical Background

As with any standard, SDTS was developed to solve a recurring problem. As the computer industry in the 1970's was maturing and developing, computers began to be used for many diverse applications including cartography, spatial data analysis, and image processing. Increasing amounts of digital spatial data were being generated and stored. Computer systems and applications that could generate and use spatial data were incompatible. Spatial data could not be reused or shared or exchanged between different computer systems. The lack of standards inhibited the data exchange of spatial data between data producers and consumers. SDTS was developed to address this issue.

The history of SDTS spans three decades. These can be characterized as the1980's are the emergence of a standard, 1990's are implementation, and the 2000's are a time of revision.

In the 1980's, the need for a data exchange standard was recognized, so the work began. The spatial data industry was in its infancy, so a common set of terms needed to be developed. The work on defining spatial objects yielded terms like "chain" and "GT-polygon". Feature sets were as varied as the data producers, so common entity names like "watercourse" were proposed. SDTS is the result of the combined efforts of the government, private industry, and the academic community. For a detailed historical account of the early development of SDTS, including the groups, individuals, methods and processes, see *The American Cartographer* January 1988.

For SDTS, the 1990's were an era of formal standardization, harmonization, profile development, promotion, and implementation. In 1992, SDTS was ratified as a NIST FIPS PUB173. ANSI deals with standards that are of broad interest in the United States, so it was a significant milestone when ANSI ratified SDTS as ANSI NCITS 320-1998. SDTS was also having an impact internationally. Australia and New Zealand adopted SDTS with some modifications in 1994 as Australian/New Zealand Standard (AS/NZS) 4270. As SDTS was proceeding as a standard, other organizations like the military were also developing standards, so various efforts to harmonize the standards were undertaken. The International Organization for Standardization (ISO) Technical Committee 211 (ISO TC 211) for Geographic Information/Geomatics began developing a suite of standards for geospatial data and processes. The United States brought SDTS forward as a contribution to this effort.

The SDTS is a very broad standard with options for all types of vector, raster, and point data. Profiles are subsets of the standard that focus on certain spatial data types and make software implementations more feasible. The first SDTS profile was the Topological Vector Profile which was ratified in 1994. USGS was named as the maintenance authority for SDTS, and as such helped coordinate the development of profiles. USGS was actively promoting

SDTS by holding implementation workshops, supporting vendors to develop import and export in their GIS software, developing public domain utilities, authoring training materials and user guides, and coordinating the articles for two special editions of *Cartography and Geographic Information Systems* in 1992 and 1994. USGS also wanted to lead by example. USGS did a mass conversion of their 1:100k scale Digital Line Graph files for the hydrography and transportation themes to the new Topological Vector Profile and then gave them away free over the Internet. This had two effects – it modernized the data, and it created a demand for software vendors to support SDTS. USGS also converted the 1:24K DLGs and the USGS DEMs to SDTS profiles throughout the last half of the 1990's.

For SDTS, the 2000's are a time of revision to survive. As with any GIS standard, SDTS is intertwined with the evolution of information technology and must change to keep up. The file encoding method used in Part 3 of SDTS (ISO 8211) is antiquated by today's IT standards. Updating SDTS to use a modern encoding such as XML or GML has been studied and is feasible, but has not had resources committed. Many other solutions to the problem of data exchange have developed which have supplanted much of the need for SDTS. However, the problem of archiving data is still largely unresolved, so perhaps that is where SDTS needs to be focused in the future.

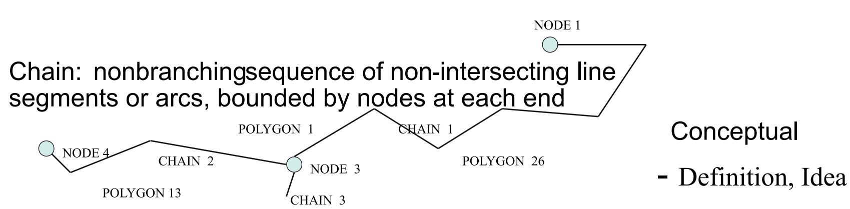
SDTS is historically significant as the first consensual GIS standard. It had to break a lot of new ground to create a mindset for the acceptance of future standards. The GIS industry no longer asks why do we need standards, but now asks which ones. SDTS has also left a lasting legacy of a vocabulary for the spatial data industry. A glossary of spatial objects, a definitive set of data quality characteristics, and the concept of self-documenting data files has had a lasting effect on the industry.

Scientific Fundamentals

The Spatial Data Transfer Standard was designed with the fundamental goal of accomplishing a transfer without losing information. A complete spatial data set is much more than a set of spatial coordinates. A complete transfer includes the entity and attributes values, but also an understanding of how to interpret them. Arrays of x,y coordinates alone do not specify the projection or datum or positional accuracy. Knowing that the data set contains hydrographic features is important, but that does not indicate the collection criteria used.

Self-contained Transfers

To ensure that all of the information needed to correctly interpret a spatial data set is available, SDTS was designed as a self-contained transfer mechanism. The set of files that constitute a transfer include the spatial coordinates, the projection, the attribute values, the attribute definitions, the positional accuracy, the collection method and other such metadata and contextual information. The philosophy of being self-contained was a new concept at the time, and was much more challenging to implement than just a file format standard.



Line	Attribute	PolyLeft	PolyRight	StartNode	EndNode	Spatial Addr
LE01#1	road	PC01 1	PC01 26	NO01 3	NOO1 1	123,421 265,526
LE01#2	road	PC01 1	PC01 13	NO01 4	NO01 3	10,425 123,421
LE01#3	.....	.....	....	.....	.....	....

Logical
- Module, Field, Subfield

004412L 0600106 2304000015000000128015LINE43043ATID42086PIDL44128PI
DR45172SNID41217ENID39258SADR38297;0000;&HY01LE01;0100;&DDF RECORD IDENTIFIER;
1600;&LINE&MODN!RCID!OBRP&(A(4),I(6),A(2));2600;&ATTRIBUTE ID&*MODN!RCID&(A(4),I(6));
1600;&POLYGON ID LEFT&MODN!RCID&(A(4),I(6));1600;&POLYGON ID RIGHT&MODN!RCID&(A(4),I(6));
1600;&STARTNODE ID&MODN!RCID&(A(4),I(6));1600;&ENDNODE ID&MODN!RCID&(A(4),I(6));
2600;&SPATIAL ADDRESS&*X!Y&((2B(32)));
00162 D 00081 220400010700LINE1307PIDL1120PIDR1131SNID1142ENID115
3SADR1764; 1;LE01 1LE;PC01 1; PC01 26;NO01 3; NO01 1;
@@@M@&@{@@&t@&@i; (spatial addresses are binary)
00162 D 00081 220400010700LINE1307PIDL1120PIDR1131SNID1142ENID115
3SADR1764; 2;LE01 2LE;PC01 1;PC01 13; NO01 4;NO01 3;
@@@@@&@c@@@M@&@{;

Format
-ISO8211 File

Spatial Data Transfer Standard (SDTS), Figure 1 SDTS Model Levels

Three Model Levels

SDTS includes three major abstraction levels – conceptual, logical, and format. Figure 1 illustrates how these three model (or abstraction) levels are used in the standard. The conceptual level includes definitions for such ideas as spatial objects, data quality, entities and attributes, and data dictionary, as described in the following subsections. The logical level is a collection of data structures and information elements called modules, records, fields, and subfields. The format level is the scheme for encoding the data into files and physical records on computer media. At the format level, there are already numerous standards in the information technology industry, so SDTS chose several to reference rather than "reinventing the wheel." The most prominent is ISO 8211 (aka FIPS PUB 123) which is documented in SDTS Part 3 on file encoding.

The SDTS document is structured to make a clear distinction between the logical and formal levels. The idea was that the logical level would remain relevant much longer than the format level. Because of the rapid advances in information technology, formats would continue to evolve and change at a faster pace than spatial data concepts or data structures. Currently, the ISO 8211 file encoding is still the only option, but XML and GML are being studied as more relevant encoding options in the future.

Spatial Objects

SDTS includes definitions of spatial objects so there is no ambiguity in the most fundamental portion of a spatial data transfer. The SDTS spatial object definitions are a lasting contribution to the spatial industry, and are widely referenced.

SDTS defines zero-, one-, two-, and some three-dimensional spatial objects. There are two other characteristics that distinguish spatial objects, that being geometry (spatial location) and topology (spatial relationships).

For zero-dimensional spatial objects, there is the geometry type which is named "point" and the topology type named "node". There are similar terms for the other dimensions. Table 1 contains all of the spatial objects defined in SDTS. Once the elemental spatial objects are defined, these can be used to define aggregate spatial objects. For example, a two-dimensional array of regularly spaced "pixels" is named a "digital image." The most common form of a vector data set is called a "two-dimensional manifold" this being a "planar graph" and its polygons.

By defining this set of spatial objects, SDTS is capable of transferring any spatial data model – vector topology, vector geometry, raster image, raster grid, and point.

Data Quality Report

Each component of SDTS was included to address a specific barrier to data transfer. One of the primary objectives of data transfer is to enable re-use beyond the original purpose and context for which the data was collected. In order to re-use data, it needs to be assessed to determine if it is suitable for the application. The information that supports suitability assessment is called the "data quality report." The data quality report contains a lot of text that is intended for human consumption.

SDTS defines five types of data quality information: lineage, positional accuracy, attribute accuracy, logical consistency, and completeness. Lineage describes the source material and derivation process. Positional accuracy refers to the horizontal and vertical accuracy and the measurement method. Attribute accuracy refers to the confidence of the attribute values and the verification method. Logical consistency refers to the correctness of spatial relationships such as valid topological relationships. Completeness refers to the selection criteria used during data collection and the method used to ensure the criteria were always followed.

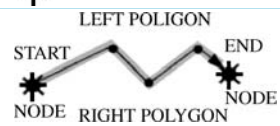
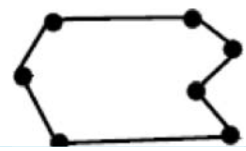
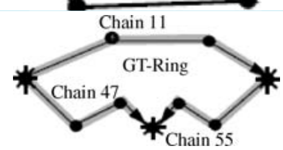
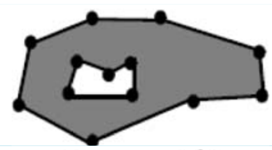
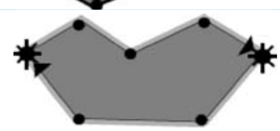
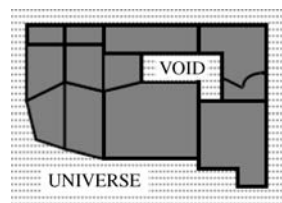
The data quality concepts constitute a major contribution to the spatial data industry, and are incorporated into the FGDC *Content Standard for Digital Geospatial Metadata.*

Feature Entities and Attributes

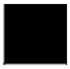
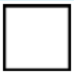
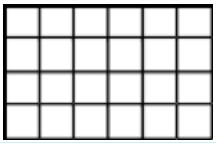
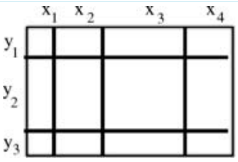
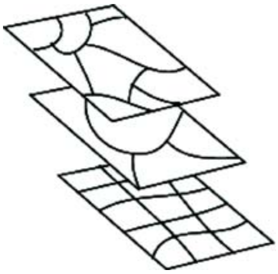
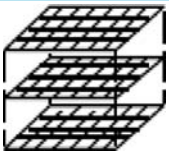
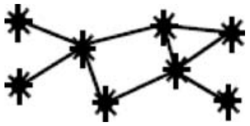
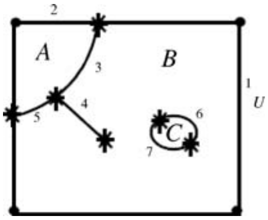
Another major barrier to data transfer is the differences in feature classification and attribution schemes. SDTS Part 2 is a standardized feature and attribute catalog of a limited set hydrographic, transportation, and other features. SDTS Part 2 represents a very unique approach to the problem of translating feature entity and attribute terminology. The SDTS Part 2 feature entity set is very different from any other feature catalog because the classification does not use a hierarchy or a specific world view.

The SDTS entities are defined by root characteristics and are mutually exclusive and are labeled with a standard term. One such standard term is "watercourse" which is defined by the root characteristic of "a way or course through which water may or does flow". The term "watercourse" also includes the terms: river, creek, stream, branch, run, kill, awawa, and canal. These included terms are listed in the standard to assist with relating common terms to the standard terms. SDTS Part 2 also contains a list of standard attributes such as "artificially improved/manmade." Many of the attribute terms are structured as to contain either yes/no values or numbers to simplify translation. The subtle variations in an entity classified as a "watercourse" can be described through the use of the standard attributes. A "river" is distinguished from

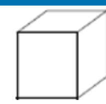
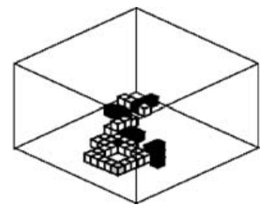
Spatial Data Transfer Standard (SDTS), Table 1 SDTS Spatial Objects

Dimension	SDTS Term and definition	Illustration	Properties
Zero	**Point.** A zero-dimensional object that specifies geometric location. One coordinate pair or triplet specifies the location. There are three sub-types: entity point, area point, and label point		Geometry; Point; Vector
Zero	**Node.** A zero-dimensional object that is a topological junction of two or more links or chains, or an end point of a link or chain		Topology; Vector
One	**Line Segment.** A direct line between two points		Geometry
One	**String.** A connected nonbranching sequence of line segments specified as the ordered sequence of points between those line segments. Note: A string may intersect itself or other strings		Geometry; Vector
One	**Arc.** A locus of points that forms a curve that is defined by a mathematical expression		Geometry; Vector
One	**Link.** A topological connection between two nodes. A link may be directed by ordering its nodes		Geometry; Topology; Vector
One	**Chain.** A directed nonbranching sequence of nonintersecting line segments and (or) arcs bounded by nodes, not necessarily distinct, at each end. There are three subtypes: complete chain, area chain, and network chain		Geometry; Topology; Vector
Two	**G-ring.** A sequence of nonintersecting strings and (or) arcs, with closure. A ring represents a closed boundary, but not the interior area inside the closed boundary		Geometry; Vector
Two	**GT-ring.** A sequence of nonintersecting complete or area chains, with closure. A ring represents a closed boundary, but not the interior area inside the closed boundary		Geometry; Topology; Vector
Two	**G-Polygon.** An area consisting of an interior area, one outer G-ring and zero or more nonintersecting, non-nested inner G-rings. No ring, inner or outer, must be collinear with or intersect any other ring of the same G-polygon		Geometry; Vector
Two	**GT-Polygon.** An area that is an atomic two-dimensional component of one and only one two-dimensional manifold. The boundary of a GT-polygon may be defined by GT-rings created from its bounding chains. A GT-polygon may also be associated with its chains (either the bounding set, or the complete set) by direct reference to these chains. The complete set of chains associated with a GT-polygon may also be found by examining the polygon references on the chains		Geometry; Topology; Vector
Two	**Universe Polygon.** Defines the part of the universe that is outside the perimeter of the area covered by other GT-polygons ("covered area") and completes the two-dimensional manifold. The boundary of the universe polygon is represented by one or more inner rings and no outer ring		Topology; Vector specifically for 2-d manifold
Two	**Void Polygon.** Defines a part of the two-dimensional manifold that is bounded by other GT-polygons, but otherwise has the same characteristics as the universe polygon. The geometry and topology of a void polygon are those of a GT-polygon	(See illustration for Universe Polygon)	Topology; Geometry; Vector specifically for 2-d manifold

Spatial Data Transfer Standard (SDTS), Table 1 (continued)

Dimension	SDTS Term and definition	Illustration	Properties
Two	**Pixel.** A two-dimensional *(geospatial)* picture element that is the smallest nondivisible element of a digital image		Geometry; Raster
Two	**Grid cell.** A two-dimensional *(geospatial)* object that represents the smallest nondivisible element of a grid		Geometry; Raster
Two	**Digital Image.** A two-dimensional *(geospatial)* array of regularly spaced picture elements (pixels) constituting a picture		Aggregate; Geometry; Raster
Two	**Grid.** A two-dimensional (geospatial) set of grid cells forming a regular tessellation of a surface		Aggregate; Geometry; Raster
Two	**Rectangle Variant Grid.** Each row and column of the grid may have an independent thickness or width		Aggregate; Geometry; Raster
Two or Three	**Layer.** An areally distributed set of spatial data representing entity instances within one theme, or having one common attribute or attribute value in an association of spatial objects. In the context of raster data, a layer is specifically a two, three or N-dimensional array of attribute values associated with all or part of a grid, image, voxel space or any other type of raster data		Aggregate; Geometry; Topology; Raster; Vector
Two or Three	**Raster.** One or more related overlapping layers for the same grid, digital image, voxel space or any other type of raster data. The corresponding cells between layers are registered to the same raster object scan reference system. The layers overlap but need not be of the same spatial extent		Aggregate; Geometry; Raster
Two	**Planar Graph.** A set of topologically interrelated zero-dimensional (node) and one-dimensional (link or chain) spatial objects, with the following rules: each link or chain is bounded by an ordered pair of nodes, not necessarily distinct; a node may bound one or more links or chains; and links or chains may only intersect at nodes. The objects can be represented as though they occur upon a planar surface		Aggregate; Topology; Geometry; Vector
Two	**Two-dimensional Manifold.** A planar graph and its associated two dimensional objects. Each chain bounds two and only two, not necessarily distinct, GT-polygons. The GT-polygons are mutually exclusive and completely exhaust the surface		Aggregate; Topology; Geometry; Vector
Two	**Network.** A graph without two dimensional objects. If projected onto a two-dimensional surface, a network can have either more than one node at a point and (or) intersecting links or chains without corresponding nodes		Aggregate; Topology; Geometry; Vector

Spatial Data Transfer Standard (SDTS), Table 1 (continued)

Dimension	SDTS Term and definition	Illustration	Properties
Three	**Voxel.** A three-dimensional (geospatial) object that represents the smallest nondivisible unit of a voxel space (volume). (The voxel is the three-dimensional conceptual equivalent of the two-dimensional pixel or grid cell.)		Simple; Geometry; Raster
Three	**Voxel Space.** A three-dimensional (geospatial) array of voxels in which the volumetric dataset (the object) resides. The volume represents some measurable properties or independent variables of a real object or phenomenon		Aggregate; Geometry; Raster

a "canal" by using the attribute "artificially improved/man-made" with a "no" value for river and a "yes" for canal.

Data Dictionary

The standardized set of feature and attribute terms was one way to solve the translation problem. If that was not feasible, SDTS allowed another option. SDTS permits a transfer to contain the definitions as provided by the data producer in a "data dictionary."

The purpose of the data dictionary component of a spatial data transfer is to convey the meaning of entities and attributes, the relationship between entities and attributes, and the domain of values for each attribute. With all of the textual definitions, a data dictionary can become large. It could be impractical to include this much extra information with every spatial data transfer from the same data producer. Therefore, SDTS permits a data dictionary only transfer.

Development of Profiles

As described under Spatial Objects, SDTS can handle a very diverse set of spatial data models. A specific instance of a spatial data transfer usually only involves a single type of spatial data like vector topology or raster image. SDTS also has many options for spatial coordinates, metadata structure, data quality, file organization, etc. The options in SDTS are overwhelming, and it is not feasible to implement them all for every specific transfer. The authors of SDTS wanted to include all the options, so the solution to controlling the complexities of implementing the standard is "profiles".

The SDTS is implemented through "profiles". Profiles are limited subsets of the standard, designed for a specific type of data. Profiles restrict options to make encoding and decoding choices and software support tools feasible.

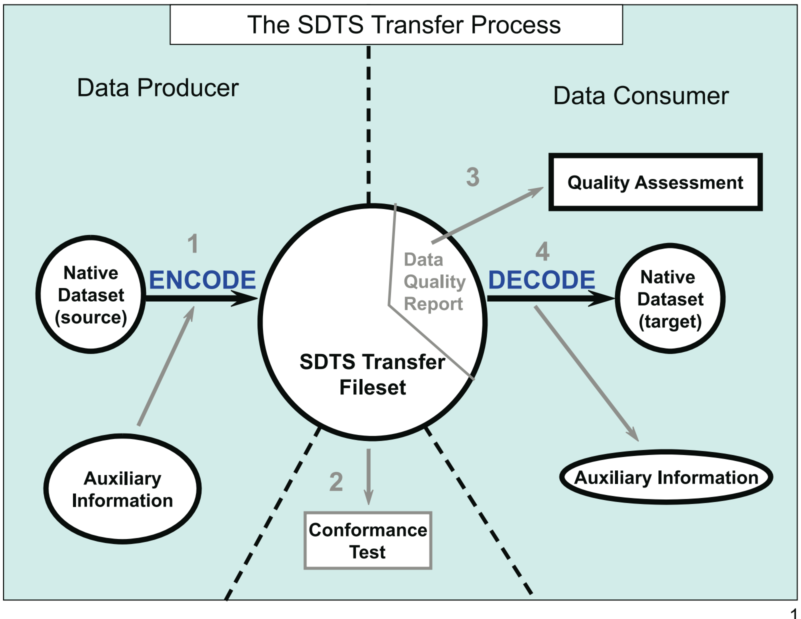
Profiles for SDTS are numbered as additional "parts" to the standard document. The base SDTS standard has three parts, so the profiles are numbered as Part 4 and so on. Each profile document parallels the SDTS document but it is not written to stand-alone. The profile can only be correctly interpreted in the context of the base SDTS standard. The first profile developed was Part 4: Topological Vector Profile (TVP) and it is currently the only profile included in the ANSI NCITS 320-1998 version of SDTS. Additional profiles are approved by the Federal Geographic Data Committee. As the first profile, the TVP paved the way for other profiles. Currently, all SDTS profiles are differentiated by spatial data model, but this is not a requirement.

Profiles are more restrictive than the base standard by limiting choices. In the TVP, the only spatial address encoding allowed is 32-bit signed integers. The spatial data must be vector and topologically structured; no geometry only data and no raster data is allowed.

Profiles deal with implementation issues, so they can also extend a standard. An extension to a standard allows a capability that is not included in the base standard. In this manner, profiles can help discover what options might need to migrate into the base standard during its mandatory review cycle. The second profile developed was for raster data and it had to add an extension. Concurrent with SDTS development in the cartographic and geographic realms, there was much progress made in commercial standardization in the image processing industry. SDTS Part 5 Raster Profile with BIIF Extension allowed the use of other image file encoding formats such as JPEG. The work on Part 5 was in parallel to the effort to gain ANSI approval for SDTS. Subsequently, there are some changes in the ANSI version of SDTS to better support raster image encoding.

Transfer Process

SDTS was created to enable spatial data exchange with no loss of information. Whether the data exchange is successful depends on the transfer process. The general steps in any transfer process are the data producer encodes a source



Spatial Data Transfer Standard (SDTS), Figure 2 SDTS Transfer Process

data set plus ancillary information into an SDTS transfer file set. The data consumer then decodes the SDTS transfer file set in the target system. Figure 2 illustrates the transfer process. The resulting SDTS transfer fileset is what could be subject to conformance testing, by either party. The data consumer may optionally perform a data assessment using the data quality report before doing a complete decode.

The encoding process is very involved and takes significant effort to specify and implement. The data producer must do model mappings at all three levels of abstraction – conceptual, logical (modules), and format (files). The data producer must find appropriate information to meet all the requirements for the data quality report, the data dictionary, and the other metadata. This ancillary information needs to be structured and managed so it can be included correctly by the software tool used to create the physical SDTS transfer file set. This is much more than a reformatting which is simply presenting the same level of information in a different physical arrangement.

The decoding process can also be more involved for the data consumer. They will need to use a mix of software tools to ensure that they view the full content of the SDTS transfer, in addition to ensuring that the spatial data and attributes are correctly imported into their GIS environment.

Further Reading

An excellent overview of SDTS can be found in the article "An Overview of FIPS 173, The Spatial Data Transfer Standard" published in *Cartography and Geographic Information Systems* December 1992 edition. The USGS SDTS web site has three user guides aimed at various audiences: The Spatial Data Transfer Standard – Senior Management Overview; The Spatial Data Transfer Standard – Guide for Technical Managers; and, The Spatial Data Transfer Standard – Handbook for Technical Staff. For a data consumer perspective on SDTS, see Chapter 15 Spatial Data Transfer Standards in the 1998 book GIS Data Conversion: Strategies, Techniques, Management. For a scientific comparison of SDTS to all other international spatial data exchange standards see section United States of America Spatial Data Transfer Standard in the book Spatial Database Transfer Standards 2: Characteristics for Assessing Standards and Full Descriptions of the National and International Standards in the World. For details on all of the modules and fields of the logical level, and the ISO 8211 encoding, see the standard document ANSI NCITS 320-1998 Part 1 and Part 3.

Key Applications

SDTS is most appropriate to be used for spatial data storage and product series distribution.

Spatial Data Storage

When SDTS was originally designed in the 1980's, it was for data exchange from a data producer to a data consumer when computers were mainframes and the media of choice was magnetic tape reels. Today, there are many formats for data exchange and even for distributed geoprocessing. The value of SDTS is in its completeness and preservation of not only data but the context and meaning of the spatial data, attributes, and it data quality. SDTS remain relevant today as a spatial data storage standard.

Spatial Data Archive

The US National Archives and Records Administration (NARA) accepts SDTS as a long term storage standard for GIS data records.

Spatial Data Distribution

The federal government has a requirement to distribute any spatial data they produce in non-proprietary formats, at a minimum. Most federal agencies that produce spatial data for distribution have an entire product series. The effort to implement any SDTS-based transfer process is spread over the many hundreds or thousands of file transfer sets, so it is feasible.

Topological Vector Profile (TVP)

The USGS did a mass conversion of its Digital Line Graph level 3 (DLG-3) to the SDTS-TVP and offers these free of charge. This was primarily done to jumpstart implementation of SDTS, because a large quantity of data, especially free, will cause the creation of software tools. Another reason was to improve on the encoding of the data and to make it easier to process, as the DLG format originated in the 1970's. When the DLG data was translated in SDTS-TVP the major-minor attribute scheme was converted into relational tables, thus modernizing the data.
The USGS National Atlas of the United States offers SDTS as a download option, in addition to proprietary options, to meet a wide variety of customer requirements.

Raster Profile with BIIF Extensions (RPBE)

The USGS did a mass conversion of its Digital Elevation Model (DEM) to the SDTS-RPBE as a way to modernize its data holdings. The USGS DEM format stored raster cells (posts) in the order they were originally collected during manual stereographic profiling which was column-major order with each column potentially having a different number of posts, and the origin was bottom left. In DEM SDTS, the grid data is a regular array with data ordered in row-major order with the origin at the top left.

Point Profile

The NOAA National Geodetic Survey distributes control points in SDTS Point Profile as an option. The original intent was to offer a GIS alternative (prior to the arrival of ESRI shapefiles) to the textual data sheet report.

Software Tools

As with any technology standard, there are a variety of commercial and free tools available for end users and developers.

Geographic Information Systems Software

In the 1990's, the SDTS was supported to some extent in every major GIS software, including ESRI ArcInfo, Erdas Imagine, Intergraph, Manifold, and GRASS.

Spatial Data Conversion & Viewing

The ever increasing variety of spatial data formats created a market need for data translators. There is software that specializes in translating, converting, and viewing spatial data in almost any format. For example, the Safe Software Feature Manipulation Engine software is one such tool that supports encoding and decoding a wide variety of spatial standards and file types, including SDTS. Global Mapper Software supports viewing and some data manipulation of a wide variety of formats, including SDTS.

Public Domain Code Libraries & Tools

As the maintenance authority for SDTS, USGS sought to ensure that public domain source code libraries were available to help support SDTS software development. There is a C++ source code library called SDTS++ that is available, and there is an ISO 8211 C library available called FIPS123, in addition to some utilities.

Future Directions

With all of the advances in technology over the last 30 years, SDTS still has a place but not for its original intent of operational data exchange. The future of SDTS is as an official archive format. As an ANSI standard, SDTS must be revisited every 5 years. In the next review, it will be decided if any of the additional profiles will be picked up as Parts. In 2003, a study determined that it would be possible to use XML or GML as a file level encoding for an SDTS transfer. FGDC, ANSI NCITS, and NARA will continue to promote the further development of SDTS as the basis for GIS archive format for government records.

Cross References

► Computing Fitness of Use of Geospatial Datasets
► Geography Markup Language (GML)
► Metadata and Interoperability, Geospatial

Recommended Reading

1. American Congress on Surveying and Mapping: The Proposed Standard for Digital Cartographic Data. Am. Cartogr. **15**(1), entire issue (1988)

2. American National Standards Institute, National Committee for Information Technology Standards: ANSI NCITS 320-1998, Spatial Data Transfer Standard Parts 1–4
3. Federal Geographic Data Committee: Components of the National Spatial Data Infrastructure. Information on Geospatial Standards. http://www.fgdc.gov, viewed on November 26 (2006)
4. Morrison, J., Wortman, K.: Special Issue. Implementing the Spatial Data Transfer Standard. Cartogr. Geogr. Inf. Syst. **19**(5), entire issue (1992)
5. Wortman, K., Buttenfield, B.: Special Issue. Current Developments and Use of the Spatial Data Transfer Standard. Cartography Geogr. Inf. Syst., **21**(3), entire issue (1994)
6. Moellering, H., Hogan, R.: Spatial Database Transfer Standards 2: Characteristics for Assessing Standards and Full Descriptions of the National and International Standards in the World. International Cartographic Association, Pergamon, Oxford (1996)
7. National Archives and Records Administration: Acceptable Archive Formats for Digital GIS Records. http://www.archives.gov/records-mgmt/initiatives/digital-geospatial-data-records.html. Accessed 26 Nov 2006
8. Altheide, P.: GIS Data Conversions. Strategies, Techniques, Management, (ed.) Pat Hohl. Chapter 15 Spatial Data Transfer Standards, pp. 318–354, Onword Press, Santa Fe, NM (1998)
9. US Geological Survey. Spatial Data Transfer Standard Informational Website. http://mcmcweb.er.usgs.gov/sdts. Accessed 26 Nov 2006

Spatial Data Types with Indeterminate Boundaries

► Objects with Broad Boundaries

Spatial Data Warehouses

► Data Warehouses and GIS
► OLAP, Spatial

Spatial DBMS

► PostGIS

Spatial Decision Making of Groups

► Participatory Planning and GIS

Spatial Decision Support System

MARTIN D. CROSSLAND
School of Business, Oral Roberts University,
Tulsa, OK, USA

Synonyms

SDSS; Geographic information systems

Definition

A *spatial decision support system* (SDSS) is a computer-based system that combines conventional data, spatially referenced data and information, and decision logic as a tool for assisting a human decision-maker. It usually includes a user interface for communicating with the decision-maker. A SDSS does not actually make a decision, but instead assists and analyzing data and presenting processed information in a form that is friendly to the decision-maker.

Main Text

Decision support systems (DSS) are increasingly being combined with *geographic information systems* (GIS) to form a hybrid type of decision support tool known as a *spatial decision support system* (SDSS). Such systems combine the data and logic of a DSS with the powerful spatial referencing and spatial analytic capabilities of a GIS to form a new system that is even more valuable than the sum of its parts.

SDSS tools can be used to assist in making effective decisions for many kinds of problem sets and information domains, including natural resource management, business, health care, emergency management, and many others.

Spatial Dependence

► Autocorrelation, Spatial
► Kriging

S

Spatial Econometric Models, Prediction

R. KELLEY PACE[1], JAMES P. LESAGE[2]
[1] Department of Finance, Louisiana State University, Baton Rouge, LA, USA
[2] Department of Finance and Economics, Texas State University – San Marcos, San Marcos, TX, USA

Synonyms

Best linear unbiased prediction; BLUP

Definition

Prediction of ex-sample spatially dependent dependent variables not only uses ex-sample independent variables in conjunction with sample estimates of the associated parameters, but also uses the sample residuals and the spatial relations between the sample observations and the ex-sample observations to produce Best Linear Unbiased Predictions (BLUP). For spatial econometric models that specify space via weight matrices, BLUP has a simple and computationally feasible form.

Historical Background

In a GIS context, there are many motivations for conducting spatial prediction. First, a need often exists for accurate prediction, as in the case of real estate where accurate valuation requires use of data from neighboring properties. Second, predictions are smoother than the underlying data, and facilitate construction of easily understood maps. Third, diagnostic maps of predictions and associated residuals can be used to assess the relation among observations and their neighbors, as well as to detect systematic patterns in residuals. Fourth, missing values often arise in spatial data samples, and gaps in the data complicate map interpretation. Accordingly, spatial prediction can fill in missing values, and aid visual interpretation.

A number of alternative spatial prediction methods exist that optimize various criterion, condition on different amounts of sample information, and vary in the amount of computational effort required. In the context of a GIS with an interactive interface, spatial prediction should occur quickly to preserve interaction with the user. Simple spatial smoothing techniques are known to meet these speed requirements. For example, averaging over the nearest neighbors or contiguous observations for each observation takes little time and results in a smoother surface or map. Nearest neighbors constitutes one form of non parametric smoother. A vast non parametric regression literature provides many alternatives. For example, geographically weighted regression (GWR) as proposed by Brunsdon, Fotheringham, and Charlton (1996) conducts a least squares regression for each observation, and more smoothing occurs when using a larger number of neighbors m as the sample in each regression. This approach employs as many as n regressions for each predicted surface, and can become computationally tedious when n and/or m becomes large, since a leave-one-out cross-validation scheme is often used to select the optimal bandwidth m which determines the amount of smoothing.[1] To avoid these problems, the focus is on parametric prediction, specifically Best Linear Unbiased Prediction (BLUP) methods (Goldberger, 1962; Robinson, 1991).

[1] Pace and LeSage (2004) combined spatial autoregressive models and GWR. Using a large bandwidth (m) for GWR resulted in smooth predictions, but led to spatially dependent residuals, whereas spatial autoregressive local estimation allowed independent selection of m for smoothness without creation of obvious spatial dependence.

Scientific Fundamentals

All forms of parametric prediction involve use of parameter estimates arising from a model in conjunction with sample data information. In the case of non-spatial models, sample data information is assumed independent over space, so the prediction for each observation requires only a single observation of the exogenous explanatory variables (given the parameters). Prediction does not directly incorporate information regarding neighboring dependent and explanatory variables which lessens data requirements and computational complexity. In contrast, spatial prediction for a single observation may require sample data information from exogenous explanatory variables associated with this observation and neighboring exogenous explanatory variables, as well as information from neighboring endogenous observations on the dependent variable. The term *exogenous* prediction will refer to spatial prediction based on parameters, spatial model relations, and sample data information involving only exogenous explanatory variables. In contrast, the term *endogenous* prediction refers to spatial prediction that includes all of the aforementioned plus sample data information on endogenous variables. Clearly, endogenous spatial prediction uses more model and sample data information than exogenous spatial prediction, which in turn uses more information than non-spatial prediction. Accordingly, endogenous spatial prediction has the potential to provide better prediction accuracy.

This discussion of spatial prediction requires some notation. Let y_1 contain n_1 known observations on the dependent variable. Let y_2 represent n_2 observations on the dependent variable which are treated as unknown. Furthermore, let X_1, X_2 represent n_1, n_2 known observations on k explanatory variables. In addition, let $y = [y_1' \quad y_2']'$, $X = [X_1' \quad X_2']'$. The split between sets 1 and 2 correspond to observations in the sample used as the basis for prediction (set 1) and the sample observations to be predicted (set 2). Let n denote the total number of locations so that $n = n_1 + n_2$. The goal is to predict the potentially unknown observations y_2 using different amounts of model and sample data information.

The simplest scenario involves predicting a dependent variable y such as house prices using a single constant exogenous variable. In this scenario, $X = \iota$, a vector of ones. A non-spatial parametric method such as ordinary least-squares would produce a constant prediction for all homes/locations equal to the mean of the sample of house prices. In contrast, endogenous spatial prediction would result in predictions that average or smooth over the sample of nearby real estate prices y to predict each home/location of interest. This type of smoothing is often desired in GIS.

A more sophisticated scenario arises when imputing missing sample data values y_2 at various locations based on the available non-missing sample data information, y_1, X_1, X_2, the model and parameters. Non-spatial parametric methods would predict the missing values using sample data values of the explanatory variables X_2 at the sites containing missing values, which need not result in predictions that smooth over space. Exogenous spatial prediction methods would use sample data on the explanatory variables X_1, X_2 from multiple locations to predict missing values, resulting in spatially smoothed predictions of y_2 at the sites containing missing values. Finally, endogenous spatial methods would use all of the above information as well as the known/non-missing sample values y_1 to produce spatially smoothed predictions at the missing value sites y_2. In the context of predicting real estate prices, non-spatial methods use only information X_2 containing things such as individual house characteristics for prediction, spatial exogenous prediction employs the own house characteristics X_2 as well as characteristics of neighboring houses X_1 to predict, and spatial endogenous prediction uses X_1, X_2 as well as neighboring house prices y_1 to predict the own house price, y_2.

Parametric prediction requires a model and our interest centers on the class of spatial autoregressive models commonly used in spatial econometrics (Anselin, 1988; LeSage and Pace, 2004), shown in (1).

$$y = \lambda W y + X\beta + \varepsilon \tag{1}$$

where β is a k by 1 vector of regression parameters, λ is a scalar parameter, ε represents n true, unobservable disturbances, and W is a n by n spatial weight matrix.

If observation (location) j affects i where $i, j = 1 \ldots n$, $W_{ij} > 0$, if $i = j$, $W_{\mathrm{ii}} = 0$, and otherwise $W_{\mathrm{ij}} = 0$. If the matrix W has row sums of unity, Wy is a smoother of y since each element of Wy is a spatial average of the observations surrounding observation i. Since $W_{\mathrm{ii}} = 0$, Wy does not use y_{i} to predict itself. Note, $W^2 y$ provides even more smoothing since this constitutes a spatial average of a spatial average. This continues for higher-order smoothing where W^{t} captures tth order neighboring relations. For some integer t, non-zero elements of $(W^t)_{ij}$ indicate observations j that are spatially connected to observation i by paths of length t. In a GIS context, W often reflects map contiguity, so information needed to form the matrix W already exists in the GIS system. When the spatial connectivity relations contained in the matrix W representadjacent or contiguous regions, or a small number of nearest neighboring observations, W would be a sparse matrix, consisting mainly of zero values. For example, using m nearest neighbors would result in the matrix W containing a proportion (m/n) of non-zero values, and this proportion becomes very small for large n. Calculating quantities such as Wy requires only order n operations when using sparse matrix techniques, and this takes little time or storage. Consequently, computations involving W require little extra expense in a GIS setting when using sparse techniques.

If $X = [\iota \quad \mathrm{U} \quad \mathrm{WU}]$ where U is some n by p matrix of underlying exogenous explanatory variables, (1) becomes the spatial Durbin model and can subsume the spatial error model and the autoregressive models. The autoregressive model results when the parameters associated with WU equal zero. The error model results when the parameters associated with WU equal $-\lambda$ times the parameter associated with U. The generality of the spatial Durbin model coupled with its ease of presentation and generality make it well-suited for many applications as discussed in LeSage and Pace (2004) and Anselin (1988).

For spatial prediction, assume that W is known or specified a priori based on the spatial configuration of the regions, each of which is associated with an observational unit in the model. In addition, assume that estimates of the model parameters β, σ_ϵ^2 and λ already exist.[2] In this model Ω is the variance-covariance matrix, and Ψ is the inverse variance-covariance or precision matrix.

$$Z = I - \lambda W \tag{2}$$

$$\Psi = Z'Z \tag{3}$$

$$\Omega = Z^{-1}(Z')^{-1} = \Psi^{-1}\,. \tag{4}$$

Using (1) to solve for y, the expectation yields a predictor $\tilde{y}$ that appears in (5).

$$E(y) = \tilde{y} = Z^{-1} X\beta \tag{5}$$

$$\tilde{e} = y - \tilde{y}\,. \tag{6}$$

This predictor uses only X in conjunction with the model parameter estimates to form the prediction $\tilde{y}$.[3] Clearly, this is exogenous spatial prediction. Let $\tilde{y} = \begin{bmatrix} \tilde{y}_1 & \tilde{y}_2 \end{bmatrix}$. Both X_1 and X_2 enter the prediction of $\tilde{y}_1$ as well as $\tilde{y}_2$ as can be

[2] For details on derivation of these estimates see LeSage and Pace (2004).

[3] The problem of finding $(I - \lambda W)^{-1} X\beta$ has been addressed in a variety of ways LeSage and Pace (2004); Pace and Barry (1997), and requires little time when using sparse weight matrices.

S

seen from expanding (5) which yields (7) and (8) (where $V = Z^{-1}$).

$$\tilde{y}_1 = V_{11}X_1\beta + V_{12}X_2\beta \tag{7}$$

$$\tilde{y}_2 = V_{21}X_1\beta + V_{22}X_2\beta \tag{8}$$

Computationally, one would never use (7) and (8) as this requires finding the inverse of Z (V in (7), (8)). This inverse is all non-zeros, and thus becomes difficult to use for large n. To see this consider the Taylor expansion of Z^{-1} (also known as the Leontief expansion),

$$Z^{-1} = I_n + \lambda W + \lambda^2 W^2 + \lambda^3 W^3 + \dots \tag{9}$$

which involves all powers of W, and for a large enough power t all observations are tth or lower order neighbors to each other.

Fortunately, one can instead solve the equation $Za = X\beta$ for a. This does not require an explicit inverse, and solving this equation can be performed very efficiently for sparse Z. Now $\tilde{y} = a$ and partitioning the n by 1 vector $\tilde{y}$ gives the desired $\tilde{y}_1$ and $\tilde{y}_2$.

Note, the exogenous spatial prediction has an associated residual when the dependent variable values are known. This appears as $\tilde{e}$ in (6), and has a similar partitioning $\tilde{e} = [\tilde{e}_1 \quad \tilde{e}_2]'$.

The gold standard for prediction of a dependent variable using both exogenous variables at all observations and conditioning on known values of the dependent variable for other dependent observations is Best Linear Unbiased Prediction (BLUP) (Goldberger, 1962). Robinson (1991) provides an illuminating discussion of the many uses of BLUP in statistics. For geostatistical models involving spatial data see Cressie (1993) for applications of BLUP. The geostatistical models usually involve specifying spatial dependence through the error process as opposed to the spatial lag of the y vector, and these "error models" take a simpler form than autoregressive models LeSage and Pace (2004). Recently, LeSage and Pace (2004) adapted the BLUP to the case of the spatial econometric model in (1) for the purpose of predicting multiple missing observations of the dependent variable y_2 that utilize sample data information contained in non-missing explanatory variables X_1, X_2, as well as non-missing dependent variables y_1. This is in contrast to the case of exogenous spatial prediction discussed above. Intuitively, conditioning on known sample data values for the dependent variable at other locations should aid prediction providing an improvement over exogenous spatial prediction. In addition to LeSage and Pace (2004), Kelejian and Prucha (2004) also develop BLUP predictors for spatial econometric models and they investigate performance of different approaches to prediction for these models with a focus on prediction of individual observations. Note, straightforward implementation of the BLUP prediction methods for spatial autoregressive models set forth in both LeSage and Pace (2004) as well as Kelejian and Prucha (2004) present computational challenges for an interactive environment of the type required for GIS. In the sequel appear a number of computationally efficient approaches to BLUP prediction and BLUP residual calculations for various cases of interest that might arise in GIS applications.

Returning to notation, partition the model in (1) into two parts, where the partitioning applies to the various square matrices such as Z that arise in relations associated with (1),

$$Z = \begin{bmatrix} I_{n_1} - \lambda W_{11} & -\lambda W_{12} \\ -\lambda W_{21} & I_{n_2} - \lambda W_{22} \end{bmatrix} \tag{10}$$

where W_{11} is n_1 by n_1, W_{12} is n_1 by n_2, W_{21} is n_2 by n_1, and W_{22} is n_2 by n_2. The non-singular matrix Z need not be symmetric, but $\Psi = Z'Z$ is symmetric and positive definite.

$$\Psi = \begin{bmatrix} \Psi_{11} & \Psi_{12} \\ \Psi_{21} & \Psi_{22} \end{bmatrix} . \tag{11}$$

Using the symmetry of Ψ, $\Psi_{12} = \Psi'_{21}$ and vice-versa. Also, Ψ_{12} is an n_1 by n_2 matrix, Ψ_{22} is n_2 by n_2, and Ψ_{11} is an n_1 by n_1 matrix.

Given the partitioned model format, BLUP for the second set of observations has the following form (LeSage and Pace, 2004),

$$\check{y}_2 = \tilde{y}_2 + \Omega_{21}(\Omega_{11})^{-1}(y_1 - \tilde{y}_1) . \tag{12}$$

Intuitively, BLUP involves a number of operations. First, it takes the residuals from using the exogenous prediction $y_1 - \tilde{y}_1$ observed on the first set of observations and removes covariance among these observations through multiplication by part of the inverse variance-covariance matrix that pertains to these observations, Ω_{11}^{-1}. Second, BLUP reimposes the pattern of dependence appropriate for the second set of data through multiplication by the partition of the variance-covariance matrix that relates the two sets of observations, Ω_{12}. The BLUP, $\check{y}_2$, then uses these new dependent residuals to improve on the exogenous prediction $\tilde{y}_2$. In other words, BLUP uses the known residuals on the first set of observations, but translates the spatial dependence structure to one appropriate for the second set of observations to be predicted.

Also, the estimated variance-covariance matrix allows BLUP to adapt to the level of spatial dependence in the sample data for any particular problem. In cases where the sample data exhibit substantial spatial dependence, BLUP predictions rely more heavily on neighboring observation values. In cases where the sample data exhibits weak dependence, BLUP can approach ordinary least-squares prediction values.

The BLUP formula (12) in its most straightforward form is not suited for the rapid computation needed for interactive GIS since it requires inversion of the precision matrix to obtain the needed components. Fortunately, via Corollary 8.5.12 in Harville (1997),

$$\Omega_{21}(\Omega_{11})^{-1} = -(\Psi_{22})^{-1}\Psi_{21} \tag{13}$$

and this relation greatly simplifies computation of the BLUP since this avoids the need to form an explicit inverse of Ψ in its entirety. Restating the BLUP with these changes yields a simple form.

$$\breve{y}_2 = \tilde{y}_2 - (\Psi_{22})^{-1}\Psi_{21}\tilde{e}_1 \,. \tag{14}$$

Given the general case, three specific cases are of interest. Specifically, the case of BLUP for spatially disconnected data, individual observation BLUP residuals, and multiple observation BLUP.

With regard to the case of BLUP for spatially disconnected data, sometimes estimates from one data sample containing n_1 observations/locations are used to make predictions on another sample containing n_2 locations. Consider the case where the two data samples do not contain common observations/locations, and neighbors to observations in each sample do not appear in the other sample. This scenario involves "spatially disconnected" sample data locations. In this case $W_{12} = 0_{n_1 x n_2}$ and $W_{21} = 0_{n_2 x n_1}$, resulting in a block-diagonal matrix W which captures the structure of spatial dependence in the autoregressive model.

In this case, Ψ, Ψ^{-1}, Z, Z^{-1} and matrices involving their products all inherit the block diagonal structure. Substituting $W_{21} = 0_{n_2 x n_1}$ into (14) yields $\breve{y}_2 = \tilde{y}_2$, where β and λ were estimated using y_1, X_1, W_{11}. This shows that the exogenous predictor (5) yields BLUP predictions when the sample data used for estimation and that used for prediction do not share common observations or neighbors (i. e., are completely spatially disconnected). Given the structure of the spatially disconnected problem, the BLUP for predicting y_2 equals $\breve{y} = (I - W_2)^{-1}X_2\beta$. In other words, for this problem endogenous and exogenous prediction yield the same result.

With regard to individual observation BLUP residuals, the use of "leave one out" residuals is common in ordinary least-squares regression for a number of purposes. A similar desire exists with spatial methods where prediction accuracy from using $n-1$ observations to predict a single excluded observation, might be used for model diagnostic purposes, or applied to each observation in turn for cross-validation. For a given set of parameters, the spatial connectivity structure embodied in the matrix W, and known explanatory variables, computing the set of "leave one out" BLUP residuals takes a surprisingly simple form.[4]

Using the BLUP formula (14) and definition of residuals (6), permit solving for the residual $\breve{e}_2$,

$$\breve{e}_2 = \tilde{e}_2 + (\Psi_{22})^{-1}\Psi_{21}\tilde{e}_1 \tag{15}$$

where (15) has some simplifying features since Ψ_{22} is a scalar, and therefore has a simple inverse. This equation can be further simplified,

$$\breve{e}_2 = (\Psi_{22})^{-1}(\Psi_{21}\tilde{e}_1 + \Psi_{22}\tilde{e}_2) \tag{16}$$

$$\breve{e}_2 = (\Psi_{22})^{-1}(\Psi_{2.}\tilde{e}) \tag{17}$$

where $\Psi_{2.}$ represents the 1 by n row corresponding to the observation of interest. Since $\Psi\tilde{e}$ gives $\Psi_{2.}\tilde{e}$ for all possible observations (all combinations of samples 1 and 2 where 2 has one observation), and since Ψ_{22} is easily computed for each observation, it is simple to compute individual BLUP residuals for each observation conditional upon all others.

Elaborating on this point, let D represent a diagonal matrix whose elements are the reciprocal of the diagonal elements of Ψ. That is, $D_{ii} = (\Psi_{ii})^{-1}$ and 0 otherwise for $i = 1 \ldots n$. Therefore, the n by 1 vector of BLUP residuals appears in $\breve{e} = D\Psi\tilde{e}$.

Computationally, this can be improved upon substantially. First, the diagonal elements of $\Psi = Z'Z$ equal $(Z \odot Z)\iota$ where $\odot$ is element wise multiplication. Let d, a n by 1 vector, represent this result. This provides the way to find the diagonal elements of Ψ without actually computing Ψ. Second, $\Psi\tilde{e}$ also equals $(Z'(Z\tilde{e}))$ and this equals $(Z'(Zy - X\beta))$ which now avoids the equation solution required for exogenous prediction. In fact, this computational short-cut is also possible for cases involving larger dense matrices. Lastly, $\breve{e} = (Z'(Zy - X\beta)) \oslash d$, where $\oslash$ represents element wise division. This results in low code complexity as this requires only a line or two of code in a high level language. Moreover, the computations require $O(n)$ operations, a desirable outcome for interactive software.

With regard to multiple observation BLUP, unlike the case of independence, missing values for a spatially dependent variable (y_2) result in a situation where the associated explanatory variables (X_2) contain useful information for both estimation and prediction. This issue was studied by LeSage and Pace (2004) who found that both EM and MCMC procedures could simultaneously handle estimation and imputation resulting in improved prediction relative to methods that simply ignored the information content of associated explanatory variables.

[4]Note, β or λ do not change across the n separate predictions. These could change as well, but the focus is on spatial prediction instead of estimation, and so assume a known, common β and λ exist.

S

Computationally, this is straightforward. Given $\tilde{e}_1$, multiply this by Ψ_{21} to form $\Psi_{21}\tilde{e}_1$ which is a n_2 by 1 vector, and this takes little time. Solve,

$$\Psi_{22}a = (\Psi_{21}\tilde{e}_1) \tag{18}$$

for a, and $a = (\Psi_{22})^{-1}\Psi_{21}\tilde{e}_1$. Finally, $\breve{y} = \tilde{y} - a$. Given an LU or Cholesky decomposition of Ψ, these operations can be reduced into even smaller, faster pieces. Although the use of more operations may seem complicated, performing more operations with simpler, sparser matrices accelerates the computations.

Key Applications

Prediction lies at the heart of many GIS applications. Examples include, but are not limited to, real estate appraisal, tax assessment, retail site selection, target marketing, political campaigns, allocation of governmental resources, transportation planning, and opinion surveys.

Future Directions

This entry demonstrated rapid computational approaches to producing Best Linear Unbiased Prediction (BLUP) for spatial autoregressive models. In contrast to non-spatial models where independence for each observation requires only a single observation of the exogenous variables for prediction, spatial prediction for a single observation may potentially involve the entire set of both explanatory as well as dependent variable sample data observations. Clarifying definitions of spatial autoregressive prediction approaches based on exogenous and endogenous sample data information were set forth in the context of BLUP.

The computational results set forth here for traditional spatial autoregressive models are also applicable to other spatial econometric approaches to modeling dependent sample data such as LeSage and Pace (2006).

A topic not covered here was estimation of the parameters β, λ, ρ in the context of missing data y_2, with y_1, X_1, X_2 known. It is possible to produce EM estimates and inferences for the model parameters that take into account all of the known sample data, and the computational results presented here are applicable in this regard. EM estimates require "fill-in" of the missing data y_2 using the conditional expectation $E(y_2)$ presented here, (the E-step) and calculation of the parameter estimates (the M-step) through optimization based on this repaired or complete sample. Continued iteration between the E- and M-steps results in estimates that take all of the known sample data information as well as the spatial dependence relations into account. The computational results set forth here allow efficient calculation of the E-step for EM estimation of the spatial autoregressive model.

It is also the case that efficient calculation of the conditional expectation $E(y_2)$ presented here would be useful for Bayesian Markov Chain Monte Carlo (MCMC) estimation of the model parameters in the context of missing data. Here, the M-step from the EM algorithm is replaced by draws from the complete sequence of conditional distributions for each parameter in the model, while the E-step remains the same as in the EM algorithm.

Cross References

► Crime Mapping and Analysis
► Decision-Making Effectiveness with GIS
► Environmental Planning and Simulation Tools
► Kriging
► Statistical Descriptions of Spatial Patterns

Recommended Reading

1. Anselin, L.: Spatial econometrics: methods and models. Kluwer, Dordrecht (1988)
2. Brunsdon, C., Fotheringham, A.S., Charlton, M.E.: Geographically Weighted Regression: A Method for Exploring Spatial Nonstationarity, Geographical Analysis, vol 28, pp. 281–298 (1996)
3. Cressie, N.: Statistics for Spatial Data. Revised edition. John Wiley, New York (1993)
4. Goldberger, A.: Best Linear Unbiased Prediction in the Linear Model, Journal of the American Statistical Association, vol. 57, pp. 369–375 (1962)
5. Harville, D.: Matrix Algebra from a Statistician's Perspective. Springer-Verlag, New York (1997)
6. Kelejian, H., Prucha, I.: Prediction Efficiencies in Spatial Models with Spatial Lags, University of Maryland (2004)
7. LeSage, J., Pace, R.K.: A Matrix Exponential Spatial Specification, Journal of Econometrics (2006)
8. LeSage, J., Pace, R.K.: Conditioning upon All the Data: Improved Prediction via Imputation, Journal of Real Estate Finance and Economics, vol. 29, pp. 233–254 (2004)
9. LeSage, J., Pace, R.K.: Introduction. In: LeSage, J.P. and Pace, R.K. (eds.) Advances in Econometrics: vol. 18: Spatial and Spatiotemporal Econometrics, pp. 1–30. Elsevier Ltd, Oxford (2004)
10. Pace, R.K., Barry, R.: Quick Computation of Spatial Autoregressive Estimators, Geographical Analysis, vol. 29, pp. 232–246 (1997)
11. Pace, R.K., LeSage, J.: Spatial Autoregressive Local Estimation. In: Mur, J., Zoller, H., Getis A. (eds.) Recent Advances in Spatial Econometrics, pp. 31–51. Palgrave Publishers (2004)
12. Robinson, G.K.: That BLUP is a Good Thing: The Estimation of Random Effects, Statistical Science, vol. 6, pp. 15–32 (1991)

Spatial Epidemiology

► Spatial Uncertainty in Medical Geography: A Geostatistical Perspective

Spatial Error Model

► Spatial Regression Models

Spatial Hydrologic Modeling

► Distributed Hydrologic Modeling

Spatial Index

► Data Collection, Reliable Real-Time

Spatial Index Structure

► R*-tree

Spatial Indexing

► Indexing, Hilbert R-tree, Spatial Indexing, Multimedia Indexing

Spatial Information

► Market and Infrastructure for Spatial Data

Spatial Information Mediation

ILYA ZASLAVSKY
San Diego Supercomputer Center,
University of California San Diego, San Diego, CA, USA

Synonyms

Wrapper-mediator approach to query evaluation

Definition

Information mediation is a query evaluation strategy where query processing against heterogeneous data sources is orchestrated by a middleware component called a *mediator*. The mediator handles query rewriting, dispatching query fragments to individual sources, and assembling individual query results into a composite query response. Source heterogeneity is managed by *source wrappers* which export a *common data model* view of the data at each source.

Main Text

Spatial information mediation extends the mediation ideas to processing complex spatial queries against sources and services that publish their spatial capabilities. This includes the ability to handle spatial data types and use spatial operations and predicates for querying spatial data sources. At the architecture level, it includes: (1) wrappers for spatial information sources that express spatial data sets using a common model, following spatial data exchange and metadata standards, (2) spatial mediators, and (3) online mapping clients that support query formulation and presentation of assembled results as maps.

Cross References

► Cyberinfrastructure for Spatial Data Integration
► Grid
► Spatial Information Mediation
► Web Services

Spatial Interaction

► Data Analysis, Spatial

Spatial Inter-Dependence

► Autocorrelation, Spatial

Spatial Interpolation

► Constraint Databases and Data Interpolation

Spatial Join

► Plane Sweep Algorithm

Spatial Lag Model

► Spatial Regression Models

Spatial Modeling Language Extension

► Modeling with Pictogrammic Languages

Spatial Models

► Public Health and Spatial Modeling

Spatial Multicriteria Decision Aid

► Multicriteria Decision Making, Spatial

Spatial Multicriteria Decision Support Systems

► Multicriteria Spatial Decision Support Systems

Spatial Multidimensional Analysis

► OLAP, Spatial

Spatial Network

► Nearest Neighbors Problem

Spatial Network Databases

► Nearest Neighbor Queries in Network Databases

Spatial Network Model

► Road Network Data Model

Spatial Objects

► Objects with Broad Boundaries

Spatial Online Analytical Processing

► OLAP, Spatial

Spatial Prediction

► Spatial and Geographically Weighted Regression

Spatial Queries

► Data Collection, Reliable Real-Time

Spatial Reference Frames

► Cadastre

Spatial Regression

► Bayesian Spatial Regression for Multi-source Predictive Mapping

Spatial Regression Models

SUMEETA SRINIVASAN
School of Engineering and Applied Sciences,
Harvard University, Cambridge, MA, USA

Synonyms

Spatial autoregressive models; Dependence, spatial; Simultaneous autoregressive model (SAR); Moving average (MA) process model; Spatial lag model; Spatial error model; Geographic weighted regression (GWR)

Definition

Spatial dependence is measured by spatial autocorrelation, which is a property of data that arises whenever there is a spatial pattern in the values located on a map, as opposed to a random pattern that indicates no spatial autocorrelation. To measure the spatial pattern (spatial association and spatial autocorrelation), some standard global and local spatial statistics have been developed. These include Moran's I, Geary's C, Getis, LISA and GLISA statistics. Besides spatial dependence in the data, there can be spatial heterogeneity. This means that the underlying process being studied may vary systematically over space. This creates problems for regression and other econometric methods that do not accommodate spatial variation in the relationships being modeled. For an ordinary least squares (OLS) estimation of the regression model the assumption is that the values of the coefficients of the independent (explanatory) variables are constant across the spatial extent of the study. If there is spatial variation in the independent variables across space then it must be confined to the error term. Also, OLS estimates are found by minimizing the sum of squared prediction errors with an assumption that error has mean zero, the error terms are uncorrelated, have a constant variance (homoskedastic) and that they are normally distributed. If there is spatial dependence amongst the data it violates these assumptions about the error term. Therefore, the standard properties of OLS do not hold and more fundamentally, a model assuming that observations are independent is incorrect. Regression models that do account for this spatial autocorrelation are called spatial regression models.

Historical Background

The theoretical work of Durbin and Watson in the 1950s on test statistics for serial autocorrelation and Imhof's approach in the 1960s to calculate the exact distribution of centrally distributed variables laid the foundation for regression models that accounted for spatial dependence [14]. In the 1970s, Cliff and Ord's key work written in 1981 on spatial autocorrelation measures with applications in regional science influenced the field of econometrics [3,4]. Research by Haining, Griffith and Anselin amongst others in developing spatial econometric models since the 1980s has resulted in making spatial regression models part of the toolkit of a variety of applied disciplines including agricultural economics and regional science. Anselin notes that, in their description of the definition of spatial econometrics, Paelinck and Klaassen stressed the importance of spatial interdependence, the asymmetry of spatial relations, and the relevance of factors located in "other spaces" [2]. Anselin suggests that the spatial econometric model-driven approach starts from a theoretical specification, which is subsequently confronted with the data [2]. He notes further that most of the methods under this category deal with estimation and specification diagnostics in spatial models [13,1,9]. Many books and monographs on spatial econometric models can be found in [1,8,10, 7].

Scientific Fundamentals

Researchers in such varied areas as climatology, agricultural sciences, ecology, public health, and urban economics need to analyze data that are multivariate, as well as spatially and temporally correlated. This motivates the need for methods that are appropriate for modeling and data analysis of complex spatial (and spatiotemporal) data sets. The spatially referenced data could be object or field data [12]. Field data (also referred to as rasters within GIS) are useful when the analysis can be assumed to represent a continuous surface as is commonly used in geostatistical analysis. Object data, which are represented as points, lines or polygons within so called vector based GIS are more typically used in econometric analysis [2].

In econometric analysis, a regression model can be written as follows:

$$Y = \beta X + \varepsilon \tag{1}$$

where Y is an $n \times 1$ matrix of the dependent variable in a data set with n data points, X are the p independent variables and an intercept β, are the $p+1$ parameters to be estimated, and ε is an $n \times 1$ random error matrix, assumed to be normally distributed in an OLS regression. If there is spatial dependence amongst the data it violates assumptions about the error term. Standard global and local spatial statistics have been developed to measure the spatial autocorrelation. Regression models that account for the spatial autocorrelation can be rewritten as

$$Y = \beta X + \Lambda L \tag{2}$$

where the matrix L is an $n \times n$ spatial filter and the product term Λ is an $n \times 1$ error matrix whose terms are not independent. Matrix L (which cannot be an identity matrix) indicates the presence of spatial autocorrelation.

Three spatial autoregressive models are commonly referred to in the literature: the conditional autoregressive model (CAR), the simultaneous autoregressive model (SAR) and the moving average (MA) process model. The main difference among them is in the variance covariance specification [9,14]. Spatial dependence in data indicates that one observation at location i depends on other observations at other locations $j \neq i$. Let the spatial arrangement of the areal units (objects) be represented by a matrix $\boldsymbol{C}$, the row-normalized version of $\boldsymbol{C}$ be $\boldsymbol{W}$ and the spatial autocorrelation parameter be ρ. The spatial contiguity matrix $\boldsymbol{W}$ is a symmetric matrix that can be generated from the topological information. According to the adjacency criteria, in the spatial weight matrix $\boldsymbol{C}$, an element c_{ij} is one if location i is adjacent to location j, and zero otherwise. Likewise, element c_{ij} can be defined to be one if the distance between location i and j is within a specified distance and zero otherwise. As the contiguity matrix does not differentiate the strength of spatial linkages between adjacent locations complex spatial weight matrices have been proposed for more precise spatial linkages. Cliff and Ord suggest a combination of distance measures and the relative length of the border between spatial units [4].

The SAR process is represented as

$$Y = \beta X + \rho W \varepsilon + \eta \tag{3}$$

$$= \beta X + \rho W (Y - \beta X) + \eta \,. \tag{4}$$

The correlated error term ε can be defined as

$$(\boldsymbol{I} - \rho \boldsymbol{W} Y)^{-1} \eta \tag{5}$$

where η is normally distributed with mean 0 and covariance $\sigma^2 \boldsymbol{I}$.

Tiefelsdorf notes that the dependent variable Y responds to (besides the error η) [14]:

i. the spatially independent exogenous component βX
ii. the spatially dependent endogenous observations $\rho \boldsymbol{W} Y$
iii. the spatial trend values $\boldsymbol{W} X \rho \beta$

S

Equation (5) leads to the formulation of the SAR model in the following manner

$$Y = \beta X + (\boldsymbol{I} - \rho \boldsymbol{W})^{-1}\eta \tag{6}$$

$$(\boldsymbol{I} - \rho \boldsymbol{W})Y = (\boldsymbol{I} - \rho \boldsymbol{W})\beta X + \eta \,. \tag{7}$$

It can be derived from (7) that the variance covariance matrix for the SAR model is

$$\sigma^2[(\boldsymbol{I} - \rho \boldsymbol{W})^{\mathrm{T}}(\boldsymbol{I} - \rho \boldsymbol{W})]^{-1} \tag{8}$$

The variance covariance matrix for a CAR model is

$$\sigma^2(\boldsymbol{I} - \rho \boldsymbol{W})^{-1} \tag{9}$$

and the inverse covariance matrix for the MA model is similar to the SAR model

$$\sigma^2(\boldsymbol{I} - \rho \boldsymbol{W})^{\mathrm{T}}(\boldsymbol{I} - \rho \boldsymbol{W}) \,. \tag{10}$$

In general, the CAR model is appropriate for situations with first order dependency or relatively local spatial autocorrelation, and SAR and MA models are more suitable where there is second order dependency or a more global spatial autocorrelation. Griffith notes that some of the advantages of the CAR model are the simple dependency structure, the minimum variance interpolations of the dependent variable, its direct extension to space time model specifications, and its ability to achieve maximum entropy among the set of all stationary models with a given finite set of variances and covariances [9]. He adds that the SAR model is analogous to the way autoregressive models are formulated for time series analysis and can be rewritten as a CAR specification, while the reverse is not necessarily true. In the case of both SAR and MA models the interpretation of ρ in matrix $\boldsymbol{W}$ is like the correlation coefficient in that it is between 0 and 1 in terms of value.

Maximum likelihood approaches have been suggested and derived for SAR and MA models by Anselin [1] and Griffith [9]. In a maximum likelihood approach, the probability of the joint distribution (likelihood) of all observations is maximized with respect to a number of relevant parameters. If the regularity conditions for the log-likelihood function are satisfied, the obtained ML estimator will achieve the desirable properties of consistency, asymptotic efficiency and asymptotic normality. Moreover, in most situations, the resulting estimates for the regular parameters of models are also unbiased [1]. Another method for estimating such models is the use of Bayesian statistics.

Two special cases of the SAR model are "spatial lag" or mixed regressive spatial autoregressive model and the "spatial error" model [1,11]. The lag model incorporate a spatially lagged dependent variable on the right hand side of the regression model [2]. Equation (3) can be written in a general form for the spatial lag model as:

$$Y = \beta X + \rho \boldsymbol{W}_1 Y + u \tag{11}$$

$$u = \lambda \boldsymbol{W}_2 u + \varepsilon \,, \tag{12}$$

ε is distributed as $N(0, \sigma^2 I)$.

For the spatial lag model specification $\boldsymbol{W}_2$ is set to 0. In other words, besides the independent explanatory variables X the spatial variation in Y is also used to predict Y. This method addresses the issue of spatial dependence but it assumes that spatial dependence only affects the dependent variables. In the case of the spatial autoregressive error model we can rewrite the OLS equation as:

$$Y = \beta X + u \tag{13}$$

$$u = \lambda \boldsymbol{W}_u + \varepsilon \,, \tag{14}$$

ε is distributed as $N(0, \sigma^2 I)$.

Here the parameter λ is a coefficient on the spatially correlated errors, similar to the serial correlated errors in time series models. The data points are considered interdependent through unmeasured variables that are correlated across space or measurement error that is correlated in space. Theoretically it is possible to eliminate this type of spatial dependence with proper explanatory variables and the correct geographical units for the data.

Geographically Weighted Regression (GWR) is another statistical technique developed by Fotheringham et al. that allows for modeling of processes that are spatially dependent [5,6]. Fotheringham et al. note that GWR results in a set of local parameter estimates for each relationship that can be mapped to produce a parameter surface across the study region and that this technique provides valuable information on the nature of the processes being investigated and supersedes traditional global types of regression modeling. GWR produces localized versions of parameter estimates as well as standard regression diagnostics including goodness-of-fit measures such as R^2.

Key Applications

Spatial regression models have applications in a number of fields and a few of these fields are further described.

Natural Sciences

In the natural sciences ecologists have been at the forefront of the use of spatial regression models to assess diversity of habitat, species patterns, vertebrate and avian species richness and environmental factors such as type of vegetation, climate and landcover type. Climate scientists have also used these models to monitor climate change and variability.

Public Health

Epidemiologists have used spatial regression models to understand the occurrence of different kinds of disease clusters and their relationship to socio-economic characteristics in the population. Air pollution studies have also used spatial regression models to link air quality and health.

Agriculture

Agricultural researchers have investigated a variety of issues using spatial regression models. These include the farm management issues such as fertilizer use and crop rotation, policy issues like land use change and relationships between livestock density and disease incidence.

Social Sciences

In the Social Sciences geographers, economists, regional scientists, demographers and political scientists have used spatial regression models. Amongst other issues, geographers have studied the incidence of homicides within cities. Demographers have used spatial regression models to forecast population in small areas. Political scientists have used it to understand such varying topics including conflict amongst countries, differences in sex ratio amongst regions and trade between countries.

Future Directions

Spatial regression models are less well developed for categorical dependent variables that are commonly used in the fields of marketing, transportation and health. The current set of models can be extended to include the case of qualitative dependent variables and also to incorporate issues of change over time and current research is addressing these issues.

Cross References

► Data Models in Commercial GIS Systems

Recommended Reading

1. Anselin, L.: Spatial Econometrics: Methods and Models. Kluwer Academic, Dordrecht (1988)
2. Anselin, L.: Under the hood Issues in the specification and interpretation of spatial regression models. Agricul. Econ. **27**(3), 247 (2002)
3. Cliff, A., Ord, J.K.: Spatial Autocorrelation, Pion, London (1973)
4. Cliff, A., Ord, J.K.: Spatial Processes, Models and Applications. Pion, London (1981)
5. Fotheringham, A.S., Brunsdon, C., Charlton, M.E.: Quantitative Geography. Sage, London (2000)
6. Fotheringham, A.S., Brunsdon, C., Charlton, M.: Geographically Weighted Regression: The Analysis of Spatially Varying Relationships. Wiley, Chichester, UK (2002)
7. Getis, A, Mur, J., Zoller, H.G. (eds.): Spatial Econometrics and Spatial Statistics. Palgrave MacMillan, New York (2004)
8. Griffith, D.: Advanced Spatial Statistics. Kluwer, Dordrecht (1988)
9. Griffith, D.A., Csillag, F.: Exploring relationships between semivariogram and spatial autoregressive models. Papers Reg. Sci. **72**(3), 283–295 (1993)
10. Haining, R.P.: Spatial Data Analysis: Theory and Practice. Cambridge University Press, Cambridge (2003)
11. LeSage, J.P.: The Theory and Practice of Spatial Econometrics. Department of Economics, University of Toledo (1999)
12. Longley, P.A., Goodchild, M.F., Maguire, D.J., Rhind, D.W.: Geographic Information Systems and Science. John Wiley & Sons, Hoboken, NJ (2005)
13. Paelinck, Jean H.P., Klaassen, L.H.: Spatial Econometric. Saxon House, England (1979)
14. Tiefelsdorf, M.: Modelling Spatial Processes – The Identification and Analysis of Spatial Relationships in Regression Residuals by Means of Moran's I. Lecture Notes in Earth Sciences, vol. 87. Springer-Verlag, Berlin (2000)

Spatial Representation of Bayesian Networks

► Bayesian Network Integration with GIS

Spatial Resolution

TEMUULEN TSAGAAN SANKEY
GIS Training and Research Center,
Idaho State University, Pocatello, ID, USA

Synonyms

Pixel size

Definition

Spatial resolution refers to the grain size or the cell size of spatial data. For example, Quickbird satellite remote sensing imagery has a resolution of 2.4 m, while AVHRR satellite imagery has a resolution of 1.1 km. Spatial resolution, therefore, determines the level of detail one can observe in a given spatial dataset.

Main Text

The terms "spatial resolution" and "scale" are sometimes confused and are used interchangeably. However, geographic scale refers to the geographic extent of the area of concern or the size of the entire area under investigation, while spatial resolution refers to the smallest measurement unit within this area. In all geographic studies and spatial analysis, the spatial resolution has to be determined based on the size of the spatial object under investigation, types of data available, and types of analyses to be conducted.

Cross References

► Scale, Effects

Spatial Road Networks

► Trip Planning Queries in Road Network Databases

Spatial Semantic Web

► Geospatial Semantic Web

Spatial Statistical Analysis

► Indoor Positioning

Spatial Statistics Program

► CrimeStat: A Spatial Statistical Program for the Analysis of Crime Incidents

Spatial Time Series

► Correlation Queries in Spatial Time Series Data

Spatial Uncertainty in Medical Geography: A Geostatistical Perspective

PIERRE GOOVAERTS
BioMedware Inc, Ann Arbor, MI, USA

Synonyms

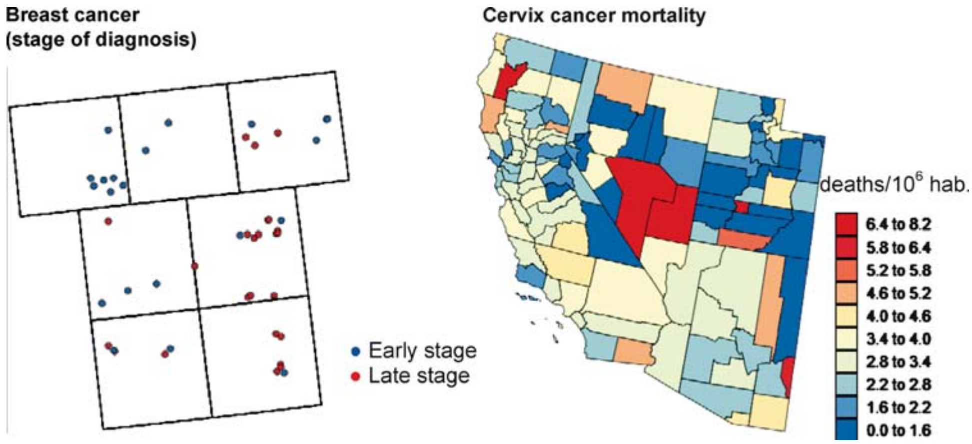
Health geographics; Spatial epidemiology; Theory of random functions; Spatial uncertainty modeling

Definition

Medical geography is defined as the branch of Human Geography concerned with the geographic aspects of health, disease and health care [1]. Data available for the study of spatial patterns of disease incidence and mortality, as well as the identification of potential causes, fall within two main categories: individual-level data or aggregated data; see Fig. 1. Although individual humans represent the basic unit of spatial analysis, case data are often aggregated to a sufficient extent to prevent the disclosure or reconstruction of patient identity [2]. The information available thus takes the form of disease rates, e. g. number of deceased or infected patients per 100,000 habitants, aggregated within areas that can span a wide range of scales, such as census units or counties. Analysis of aggregated data is frequently hampered by the presence of noise caused by unreliable extreme rates computed from sparsely populated geographical entities or for diseases with a low frequency of occurrence. Ignoring the uncertainty attached to rate estimates can lead to misallocation of resources to investigate unreliable clusters of high risk, while areas of real concern might go undetected. Smoothing methods have been developed to improve the reliability of these estimates by borrowing information from neighboring entities [3]. Etymologically, the term "geostatistics" designates the statistical study of natural phenomena. It has recently been extended to health science to incorporate the size and shape of administrative units, as well as the spatial dependence of data, into the filtering of noisy rates and the quantification of the corresponding uncertainty [4,5].

Historical Background

The idea that place and location can influence health is a very old and familiar concept in medical geography. One of the first demonstrations of the power of mapping and analyzing health data was provided by Dr. John Snow's study of the cholera epidemic that ravaged London in 1854. Using maps showing the locations of water pumps and the homes of people who died of cholera, Snow was able to deduce that one public pump was the source of the cholera outbreak [6]. Since then, the field of medical geography has come a long way, replacing paper maps with digital maps in what is now called Geographic Information Systems (GIS). Similarly, descriptive speculation about disease has given place to scientific analysis of spatial patterns of disease including hypothesis testing, multilevel modeling, regression and multivariate analysis [3].
Notwithstanding the contributions of many others, including Gandin, Matern, Yaglom, or Krige, Dr. Georges Matheron formalized the discipline of geostatistics as it is known today. The early developments of geostatistics in the 1960s' aimed to improve the evaluation of recoverable reserves in mining deposits [7]. Its field of application expanded considerably to encompass nowadays most fields of geoscience (e. g. geology, geochemistry, geohydrology, soil science) and a vast array of disciplines that all deal with the analysis of space-time data, such as



Spatial Uncertainty in Medical Geography: A Geostatistical Perspective, Figure 1 Illustration of the two main types of data available for human health studies: individual-level data (residences of patients with early or late stage diagnosis for breast cancer) and county-level mortality rates for cervix cancer

oceanography, hydrogeology, remote sensing, agriculture, and environmental sciences. Yet, the use of geostatistics in medical geography is still in its infancy. Transferring to health science methods originally developed for the analysis of earth properties presents several methodological and technical challenges that arise from the fact that health data are typically aggregated over irregular spatial supports and consist of a numerator and a denominator (i. e. rates).

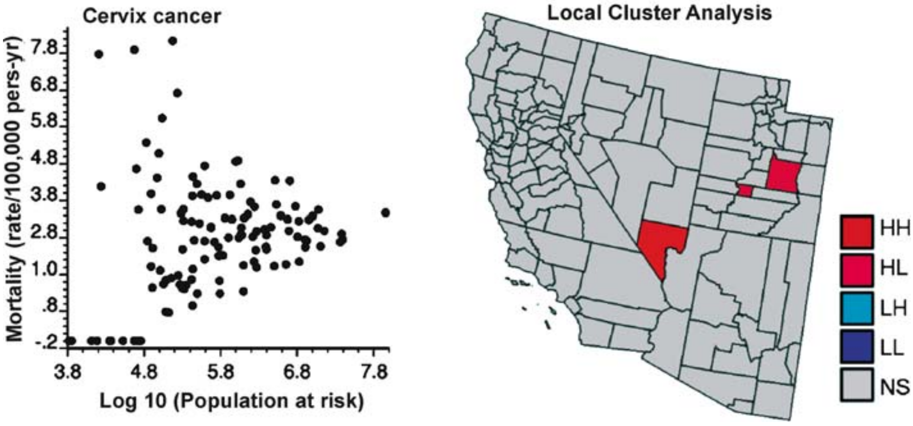
The first initiative to tailor geostatistical tools to the analysis of disease rates must be credited to Oliver et al.'s study on the risk of childhood cancer in the West Midlands of England [8]. A similar approach, called Poisson kriging, was developed more recently in the field of marine ecology and generalized to the analysis of cancer mortality rates and cholera incidence data [5,9,10]. Poisson kriging was combined with stochastic simulation to generate multiple realizations of the spatial distribution of cancer mortality risk, allowing the propagation of uncertainty through the detection of cancer clusters and outliers [11]. Complex random field models that require distributional assumptions and computationally intensive parameter estimation using Markov Chain Monte Carlo (MCMC) techniques were also developed [12,13]. A limitation of all these studies is the assumption that the size and shape of geographical units, as well as the distribution of the population within those units, are uniform, which is often inappropriate. The critical issue of change of support has just started being addressed in the literature. For example, geostatistics was used for mapping the number of low birth weight (LBW) babies at the Census tract level, accounting for county-level LBW data and covariates measured over different spatial supports, such as a fine grid of ground-level particulate matter concentrations or tract population [14].

Scientific Fundamentals

A visual inspection of the cervix cancer mortality map in Fig. 1 conveys the impression that rates are particularly high in the centre of the study area, as well as in a few Northern California counties. This result must, however, be interpreted with caution since the population is not uniformly distributed across the study area and rates computed from sparsely populated counties tend to be less reliable. This effect, known as "small number problem", is illustrated by the scattergram in Fig. 2. In particular, cervix cancer mortality in excess of 5 deaths per 100,000 person-years is observed only for counties with a 25 year cumulated population smaller than 10^5 habitants. Ignoring the small number problem might lead to spurious conclusions when investigating the existence of local clusters and outliers of high or low cancer risk values. For example, Fig. 2 (right map) shows the results of a local cluster analysis [11] of the mortality map. Two counties are declared significant High-Low (HL) outliers (i. e. high mortality rate surrounded by counties with a low averaged rate), a result that must be interpreted with caution given their small population sizes. The objective of the geostatistical analysis is thus to obtain more reliable estimates of the mortality rates and propagate the uncertainty attached to the cancer mortality map through its analysis, such as the detection of counties that depart significantly from the trend displayed across the region.

For a given number N of geographical units v_α (e. g. counties), denote the observed mortality rates (areal data) as $z(v_\alpha) = d(v_\alpha) / n(v_\alpha)$, where $d(v_\alpha)$ is the number of recorded mortality cases and $n(v_\alpha)$ is the size of the population at risk. Smoothing methods have been developed to improve the reliability of observed rates by borrowing informa-

S



Spatial Uncertainty in Medical Geography: A Geostatistical Perspective, Figure 2 Scatterplot illustrates the larger variance of rates computed from sparsely populated counties (small number problem), while the local cluster analysis of the mortality map of Fig. 1 highlights the two counties that are declared significant high-low (HL) outliers and the county that is the center of a significant cluster of high rates. Most counties are found non-significant (NS)

tion from neighboring entities [3]. These methods range from simple deterministic techniques to sophisticated full Bayesian models. Geostatistics provides a model-based approach with intermediate difficulty in terms of implementation and computer requirements. The noise-filtered mortality rate for a given area v_α, called mortality risk, is estimated as a linear combination of the kernel rate $z(v_\alpha)$ and the rates observed in $(K-1)$ neighboring entities v_i:

$$\hat{r}(v_\alpha) = \sum_{i=1}^{K} \lambda_i z(v_i)\,. \tag{1}$$

The weights λ_i assigned to the K rates are computed by solving the following system of linear equations; known as "Poisson kriging" system:

$$\begin{aligned} &\sum_{j=1}^{K} \lambda_j \left[\bar{C}_R(v_i, v_j) + \delta_{ij} \frac{m*}{n(v_i)} \right] + \mu(v_\alpha) = \bar{C}_R(v_i, v_\alpha) \\ &\qquad\qquad\qquad\qquad i = 1, \ldots, K \\ &\sum_{j=1}^{K} \lambda_j = 1\,, \end{aligned} \tag{2}$$

where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise, and $m*$ is the population-weighted mean of the N rates. The "error variance" term, $m*/n(v_i)$, leads to smaller weights for less reliable data (i.e. rates measured over smaller populations). In addition to the population size, the kriging system accounts for the spatial correlation among geographical units through the area-to-area covariance terms $\bar{C}_R(v_i, v_j) = \mathrm{Cov}\{Z(v_i), Z(v_j)\}$ and $\bar{C}_R(v_i, v_\alpha)$. Those covariances are numerically approximated by averaging the point-support covariance $C(\mathbf{h})$ computed between any two locations discretizing the areas v_i and v_j:

$$\bar{C}_R(v_i, v_j) = \frac{1}{\sum_{s=1}^{P_i} \sum_{s'=1}^{P_j} w_{ss'}} \sum_{s=1}^{P_i} \sum_{s'=1}^{P_j} w_{ss'} C(\mathbf{u}_s, \mathbf{u}_{s'}) \tag{3}$$

where P_i and P_j are the number of points used to discretize the two areas v_i and v_j, respectively. The weights $w_{ss'}$ are computed as the product of population sizes assigned to each discretizing point $\mathbf{u}_s$ and $\mathbf{u}_{s'}$:

$$\begin{aligned} &w_{ss'} = n(\mathbf{u}_s) \times n(\mathbf{u}_{s'}) \\ &\text{with} \quad \sum_{s=1}^{P_i} n(\mathbf{u}_s) = n(v_i) \quad \text{and} \quad \sum_{s'=1}^{P_j} n(\mathbf{u}_{s'}) = n(v_j)\,. \end{aligned}$$

The point-support covariance of the risk $C(\mathbf{h})$, or equivalently the point-support semivariogram $\gamma(\mathbf{h})$, cannot be estimated directly from the observed rates, since only areal data are available. Thus, only the regularized semivariogram of the risk can be estimated as:

$$\begin{aligned} \hat{\gamma}_R(\mathbf{h}) = &\frac{1}{2 \sum_{\alpha,\beta}^{N(\mathbf{h})} \frac{n(v_\alpha) n(v_\beta)}{n(v_\alpha)+n(v_\beta)}} \\ &\cdot \sum_{\alpha,\beta}^{N(\mathbf{h})} \left\{ \frac{n(v_\alpha) n(v_\beta)}{n(v_\alpha) + n(v_\beta)} \left[z(v_\alpha) - z(v_\beta) \right]^2 - m* \right\} \end{aligned} \tag{4}$$

where $N(\mathbf{h})$ is the number of pairs of areas (v_α, v_β) whose population-weighted centroids are separated by the vector $\mathbf{h}$. The different spatial increments $[z(v_\alpha) - z(v_\beta)]^2$ are weighted by a function of their respective population sizes, $n(v_\alpha)n(v_\beta)/[n(v_\alpha) + n(v_\beta)]$, a term which is inversely proportional to their standard deviations. More importance is thus given to the more reliable data pairs (i.e. smaller standard deviations). Derivation of a point-support semivariogram from the experimental semivariogram $\hat{\gamma}_R(\mathbf{h})$ computed from areal data is called "deconvolution", an operation that is conducted using an iterative procedure [15].

The uncertainty about the cancer mortality risk prevailing within the geographical unit v_α can be modeled using the conditional cumulative distribution function (ccdf) of the risk variable $R(v_\alpha)$. Under the assumption of normality of the prediction errors, that ccdf is defined as:

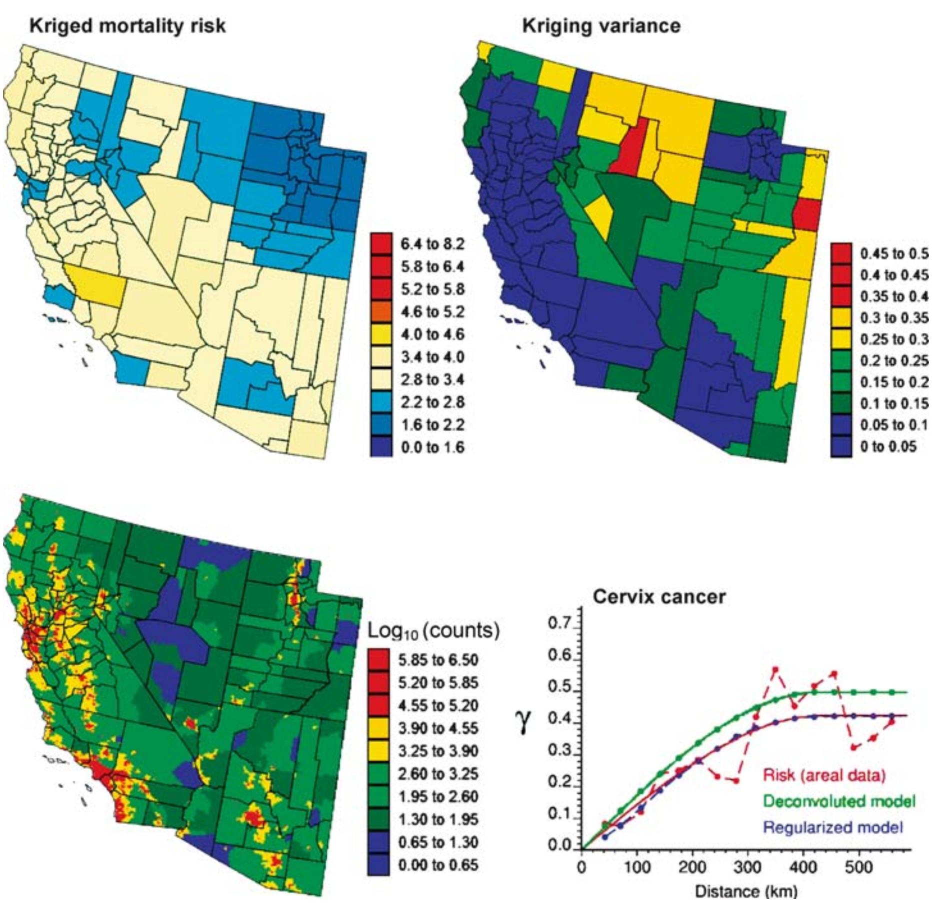
$$F(v_\alpha; r|(K)) = \Pr\text{ob}\,\{R(v_\alpha) \leqslant r|(K)\} = G\left(\frac{r - \hat{r}(v_\alpha)}{\sigma(v_\alpha)}\right) \quad (5)$$

$G(.)$ is the cumulative distribution function of the standard normal random variable, and $\sigma\,(v_\alpha)$ is the square root of the kriging variance estimated as:

$$\sigma^2(v_\alpha) = \bar{C}_R(v_\alpha, v_\alpha) - \sum_{i=1}^{K} \lambda_i \bar{C}_R(v_i, v_\alpha) - \mu(v_\alpha) \quad (6)$$

where $\bar{C}_R(v_\alpha, v_\alpha)$ is the within-area covariance that is computed according to (3) with $v_i = v_j = v_\alpha$. The notation "$|(K)$" expresses conditioning to the local information, say, K neighboring observed rates. The function (5) gives the probability that the unknown risk is no greater than any given threshold r. It is modeled as a Gaussian distribution with the mean and variance corresponding to the Poisson kriging estimate and variance. Figure 3 (top graphs) shows the maps of cervix cancer mortality risk and the associated variance estimated using Poisson kriging, accounting for the population density map and the deconvoluted point-support semivariogram model displayed at the bottom of that figure. The kriged risk map is much smoother than the original map of observed rates since the noise due to small population sizes is filtered. In particular, the high risk area formed by two central counties in Fig. 1 disappeared, which illustrates the danger of interpreting maps of observed rates. The map of kriging variance essentially reflects the higher confidence in the mortality risk estimated for counties with large populations.

A major weakness of the uncertainty measure reported in Fig. 3 is that it is area-specific; that is, it informs the uncertainty prevailing over a single geographical entity (e. g. county) at a time. Unless risk values are spatially independent, the probabilities of excess of a specific threshold computed for several entities do not provide a measure of uncertainty about whether these entities jointly exceed that threshold. In addition to a measure of "local" uncertainty, one thus needs to assess the "spatial" uncertainty, that is



Spatial Uncertainty in Medical Geography: A Geostatistical Perspective, Figure 3 Maps of cervix cancer mortality risk (deaths/100,000 habitants) and associated prediction variance estimated at the county level using Poisson kriging. *Bottom graphs* show the high-resolution population map and the deconvoluted semivariogram model used in the estimation the uncertainty attached to the spatial distribution of risk values across the study area. The quantification of spatial uncertainty is particularly important for cluster detection, since the focus is on risk values for a group of geographical entities considered simultaneously. This information is not conveyed by a single map of the estimated risk and the associated prediction variance.

Spatial uncertainty modeling has been one of the most vibrant areas of research in geostatistics for the last 2 decades [4]. Applications, such as modeling the migration of pollutants in the subsurface environment, require measures of multiple-point uncertainty, such as the probability of occurrence of a string of high or low permeability values that may represent a flow path or flow barrier. These joint probabilities are assessed numerically from a set of realizations of the spatial distribution of attribute values over the locations of interest. In other words, the spatial uncertainty is modeled through the generation of a set of equally-probable simulated maps, each consistent with the information available, such as histogram or a spatial correlation function. Stochastic simulation allows generation of maps that reproduce the spatial variability of the data without smoothing effect. The set of simulated maps can also be used to propagate the uncertainty through spatial operators or transfer functions; for example the local cluster analysis of cancer mortality maps.

Simulation of spatial phenomena can be accomplished using a growing variety of techniques that differ in the underlying random function model, the amount and type of information that can be accounted for, and the computer requirements [4]. The main difficulty for the current application is that there is no measured risk data; only a semivariogram model computed through the deconvolution of estimator (4) is available. The lack of target histogram is not an issue for the p-field simulation approach [11]. The basic idea is to generate a realization $\{r^{(l)}(v_\alpha), \alpha = 1,\ldots,N\}$ through the sampling of the set of ccdfs (5) by a set of spatially correlated probability values $\{p^{(l)}(v_\alpha), \alpha = 1,\ldots,N\}$, known as a probability field or p-field. Since the probability distributions are Gaussian, each risk value is simulated as:

$$r^{(l)}(v_\alpha) = \hat{r}(v_\alpha) + \sigma(v_\alpha) y^{(l)}(v_\alpha) \tag{7}$$

where $y^{(l)}(v_\alpha)$ is the quantile of the standard normal distribution corresponding to the cumulative probability $p^{(l)}(v^\alpha)$. The L sets of random deviates or normal scores, $\{y^{(l)}(v_\alpha), \alpha = 1,\ldots,N\}$, are generated using non-conditional sequential Gaussian simulation with semivariogram of the risk, $\gamma_R(\mathbf{h})$, rescaled to a unit sill [11].

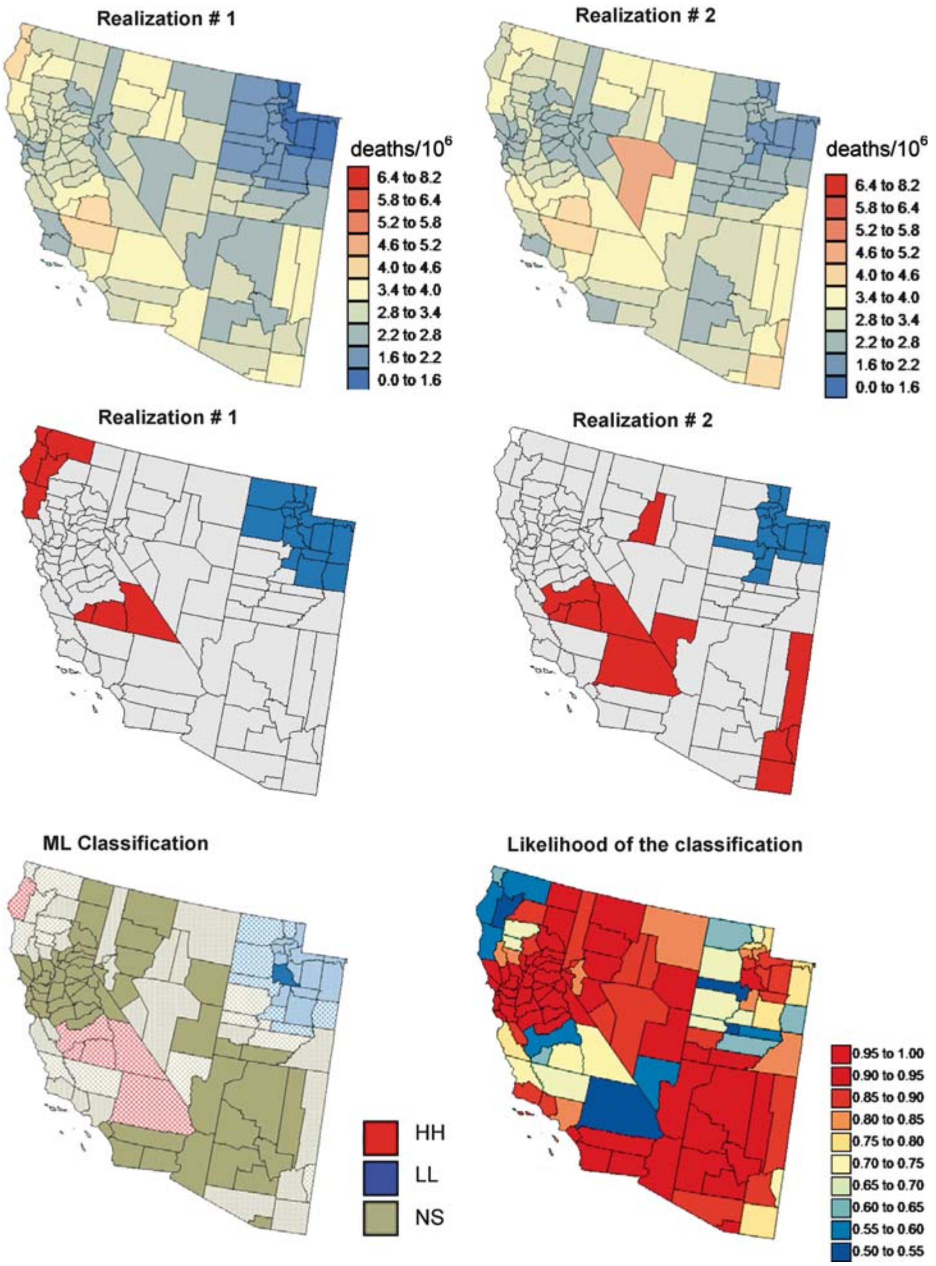
Figure 4 (top maps) shows two realizations of the spatial distribution of cervix cancer mortality risk values generated using p-field simulation. The simulated maps are more variable than the kriged risk map of Fig. 3, yet they are smoother than the map of potentially unreliable rates of Fig. 1. Differences among realizations depict the uncertainty attached to the risk map. For example, Nye County in the center of the map, which has a very high mortality rate (recall Fig. 1) but low population, has a simulated risk that is small for realization #1 but large in the next realization. Differences among simulated risk maps translate into differences among results of the local cluster analysis (middle maps in Fig. 4). Five hundreds realizations were generated and underwent a local cluster analysis (LCA). The information provided by the set of 500 LCAs is summarized at the bottom of Fig. 4. The color code indicates the most frequent classification (maximum likelihood = ML) of each county across the 500 simulated maps. The shading reflects the probability of occurrence or likelihood of the mapped class, see Fig. 4 (right bottom graph). Solid shading corresponds to classifications with high frequencies of occurrence (i. e. likelihood > 0.9), while hatched counties denote the least reliable results (i. e. likelihood < 0.75). This coding is somewhat subjective but leads to a clear visualization of the lower reliability of the clusters of high values relatively to the cluster of low risk identified in Utah. Only one county south of Salt Lake City is declared a significant low-risk cluster with a high likelihood (0.906). Thus, the two bottom maps in Fig. 4 allow one to quantify numerically how the uncertainty about the spatial distribution of health outcomes, as modelled by the set of 500 simulated risk maps, translates into uncertainty about the location of clusters of low or high values.

Key Applications

Poisson kriging has potential applications in any field that is concerned with the spatial analysis of rate data; for example: ecology, criminology, demography or geomarketing.

Ecology

Poisson kriging was originally developed for mapping the relative abundance of fin whales in the Mediterranean sea [9]. In this particular application, the numerator was the observed count while the denominator was the observational effort (i. e. total number of hours of observation). A similar approach could be used, for example, to map fish abundance from catch rates or West Nile Virus infection rates from mosquito trap data.



Spatial Uncertainty in Medical Geography: A Geostatistical Perspective, Figure 4 Two realizations of the spatial distribution of cervix cancer risk (*top row*) and the results of the local cluster analysis (*middle row*). The *left bottom map* shows the most likely (ML) classification inferred from 500 realizations. The intensity of the shading increases as the classification becomes more certain (i. e. the likelihood increases)

S

Criminology

Geographical information systems have been used since the early 1990s to assist in identifying geographical clusters of crime, commonly known as hot-spots. For confidentiality reasons, crime data (e. g. homicide, burglary) are commonly aggregated within small census geographies, and crime rates (e. g. counts per thousands households) are computed to display the level of risk and thus inform deployment of crime reduction resources. The analysis of rare events such as homicide in small populations can be greatly influenced by the small number problem and requires the application of spatial smoothers.

Demography

To explain spatial patterns in health outcomes, incidence or mortality rates are frequently correlated with sociodemographic variables measured over the same geographical units. Many of these covariates (e. g. proportion of households below the poverty level or without health insurance, unemployment rate, fraction of population over 65 year of age) are computed as ratio and so can exhibit the same noise problem as health data.

Geomarketing

The term geomarketing designates the planning, coordination and controlling of customer-oriented market activities with a GIS. Customer data, such as the number of shipped packages or TV sets per households, are often aggregated over small census geographies, which makes them susceptible to the small number problem.

Future Directions

The major difficulty in the analysis of health outcomes is that the patterns observed reflect the influence of a complex constellation of demographic, social, economic, cultural and environmental factors that likely change through time and space, and interact with the different types and scales of places where people live. It is thus primordial to incorporate the scale and spatial support of the data in their processing, as well as to account for the impact of population sizes on the reliability of rate estimates. Geostatistics provides a methodology to model the spatial correlation among rates measured over irregular geographic supports and to compute noise-free risk estimates over the same units or at much finer scales [15]. It also enables the propagation of rate uncertainty through the delineation of areas with significantly higher/lower mortality or incidence rates, as well as the analysis of relationships between health outcomes and putative risk factors. The implementation of this new methodology in user-friendly software should facilitate its adoption by the epidemiologists and GIS specialists working in health departments and cancer registries. The "Key applications" section also emphasized the wide range of applications that will benefit from further developments of Poisson kriging. In the future, the approach should be generalized to the multivariate case to analyze jointly multiple diseases or the rates of the same disease recorded for different categories of individuals (e. g. different genders or ethnic groups). Analysis of spatial relationships among diseases should facilitate the identification of common stressors, such as poverty level, lack of access to health care or environmental pollution. A multivariate approach would also enable the mapping and detection of health disparities, such as the delineation of areas where cancer mortality rates are significantly higher for minority groups. Another avenue of research is the incorporation of the temporal dimensions into the analysis. The study of temporal changes in spatial patterns would provide useful information for cancer control strategies, for example through the identification of areas where current prevention (e. g. screening for cancers) is deficient.

Cross References

► Spatial and Geographically Weighted Regression

Recommended Reading

1. May, J.M.: Medical Geography: Its Methods and Objectives. Geogr. Rev. **40**, 9–41 (1950)
2. Boscoe, F.P., Ward, M.H., Reynolds, P.: Current practices in spatial analysis of cancer data: data characteristics and data sources for geographic studies of cancer. Int. J. Health Geogr. **3**, article 28 (2004)
3. Waller, L.A., Gotway, C.A.: Applied Spatial Statistics for Public Health Data. John Wiley and Sons, New Jersey (2004)
4. Chiles, J.P.: Delfiner, P.: Geostatistics: Modeling Spatial Uncertainty. John Wiley and Sons, New York (1999)
5. Goovaerts, P.: Geostatistical analysis of disease data: estimation of cancer mortality risk from empirical frequencies using Poisson kriging. Int. J. Health Geogr. **4**, article 31 (2005)
6. McLeod K.S.: Our sense of Snow: the myth of John Snow in medical geography. Soc. Sci. & Medicine **50**, 923–935 (2000)
7. Journel, A.G., Huijbregts, C.J.: Mining Geostatistics. Academic Press, New York (1992)
8. Oliver, M.A., Webster, R., Lajaunie, C., Muir, K.R., Parkes, S.E., Cameron, A.H., Stevens, M.C.G., Mann, J.R.: Binomial cokriging for estimating and mapping the risk of childhood cancer. IMA J. Math. Appl. Med. Biol. **15**, 279–297 (1998)
9. Monestiez, P., Dubroca, L., Bonnin, E., Durbec, J.P., Guinet, C.: Geostatistical modelling of spatial distribution of Balenoptera physalus in the northwestern Mediterranean Sea from sparse count data and heterogeneous observation efforts. Ecol. Model. **193**, 615–628 (2006)
10. Ali, M., Goovaerts, P., Nazia, N., Haq, M.Z., Yunus, M., Emch, M.: Application of Poisson kriging to the mapping of cholera and dysentery incidence in an endemic area of Bangladesh. Int. J. Health Geogr., **5**, article 45 (2006)
11. Goovaerts, P.: Geostatistical analysis of disease data: visualization and propagation of spatial uncertainty in cancer mortality risk using Poisson kriging and p-field simulation. Int. J. Health Geogr. **5**, article 7 (2006)
12. Kelsall, J., Wakefield, J.: Modeling spatial variation in disease risk: a geostatistical approach. J. Am. Stat. Assoc. **97**, 692–701 (2002)
13. Choi, K.M., Serre, M.L., Christakos, G.: Efficient mapping of California mortality fields at different spatial scales. J. Expo. Anal. Environ. Epidemiol. **13**, 120–133 (2003)
14. Gotway, C.A., Young, L.J.: A geostatistical approach to linking geographically-aggregated data from different sources. Technical report #2004–012. Department of Statistics, University of Florida (2004)
15. Goovaerts, P.: Geostatistical analysis of disease data: accounting for spatial support and population density in the isopleth mapping of cancer mortality risk using area-to-point Poisson kriging. Int. J. Health Geogr. **5**, article 52 (2006)

Spatial Uncertainty Modeling

► Spatial Uncertainty in Medical Geography: A Geostatistical Perspective

Spatial Web Services

► Web Services, Geospatial

Spatial Weight Matrices

► Spatial Contiguity Matrices

Spatial Weighting Matrix

► Spatial Weights Matrix

Spatial Weights Matrix

XIAOBO ZHOU, HENRY LIN
Department of Crop and Soil Sciences, The Pennsylvania State University, University Park, PA, USA

Synonyms

Spatial weighting matrix; Proximity matrix; Continuity matrix

Definition

A spatial weights matrix is an $n \times n$ positive symmetric matrix W with element w_{ij} at location i, j for n locations. The values of w_{ij} or the weights for each pair of locations are assigned by some preset rules which define the spatial relations among locations and, therefore, determine the spatial autocorrelation statistics. By convention, $w_{ii} = 0$ for the diagonal elements.

Main Text

There are many ways to define the weights, including contiguity weights, distance weights, and other weights. The simplest form of weights is binary weight where $w_{ij} = 1$ for neighboring locations i and j, or when the distance between location i and j is less than a fixed distance, or for the fixed number of locations nearest to location i. Otherwise, $w_{ij} = 0$. Another widely used weight is the distance inverse weight in which $w_{ij} = 1/\mathrm{d}_{ij}{}^{\alpha}$. In this continuous weighting function, the changing rate of weights over distance d_{ij} is determined by the parameter α. Other more complicated spatial weights matrices are also used in calculating spatial autocorrelation statistics.

Cross References

► Autocorrelation, Spatial
► Geary's *C*
► Moran's *I*

Spatially Agglomerative Clustering

► Geodemographic Segmentation

Spatially Aware Role-Based Access Control

► Security Models, Geospatial

Spatially-Enabled Data Warehouses

► Data Warehouses and GIS

Spatio-temporal Association Patterns

► Patterns in Spatio-temporal Data

Spatio-temporal Data Mining

► Stream Mining, Spatial

Spatio-temporal Data Models

► Spatio-temporal Query Languages

Spatio-temporal Data Types

RALF HARTMUT GÜTING
Faculty for Mathematics and Computer Science, Distance University of Hagen, Hagen, Germany

Synonyms

Data types for moving objects

Definition

Abstract data types to represent time dependent geometries, in particular continuously changing geometries, or *moving objects*. The most important types are *moving point* and *moving region*.

Main Text

A *moving point* represents an entity for which only the time dependent position is of interest. A *moving region* describes an entity for which the time dependent location as well as the shape and extent are relevant. For example, moving points could represent people, vehicles, such as cars, trucks, ships or airplanes, or animals; moving regions could be hurricanes, forest fires, spread of epidemic diseases, etc. Moving point data may be captured by GPS devices or RFID tags; moving region data may result from processing sequences of satellite images, for example. Geometrically, moving points or moving regions exist in a 3D (2D + time) space if the movement is modeled within the 2D plane; for moving points this can be easily extended to 4D (3D + time).

Beyond the most relevant types moving point and moving region, to obtain a closed system there are related time dependent data types such as real-valued functions or time dependent boolean values. To have a uniform terminology these types are also called *moving real* and *moving bool*, respectively. Static spatial data types such as *point*, *line* or *region*, and standard data types are also needed. The data types include suitable operations such as

trajectory : *mpoint* → *line*

Projection of a moving point into the plane

inside : *mpoint* × *mregion* → *mbool*

When is a moving point inside a moving region

distance : *mpoint* × *point* → *mreal*

Distance between a moving and a static point .

Cross References

► Moving Object Languages

Recommended Reading

1. Erwig, M., Güting, R.H., Schneider, M., Vazirgiannis, M.: Spatio-temporal Data Types: An approach to modeling and querying moving objects in databases. GeoInformatica **3**, 265–291 (1999)

Spatio-temporal Data Warehousing

► Dynamic Travel Time Maps

Spatio-temporal Database

Xiaopeng Xiong[1], Mohamed F. Mokbel[2], Walid G. Aref[1]
[1] Department of Computer Science, Purdue University, Indianapolis, IN, USA
[2] Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA

Definition

A spatiotemporal database is a new type of database system that manages *spatiotemporal objects* and supports corresponding query functionalities. A *spatiotemporal object* is a kind of object that dynamically updates spatial locations and/or extents along with time. A typical example of a spatiotemporal object is a moving object (e. g., a car, a flight or a pedestrian) whose location continuously changes.

Spatiotemporal databases have many important applications such as Geographic Information Systems, Location-aware Systems, Traffic Monitoring Systems, and Environmental Information Systems. Due to their importance, spatiotemporal database systems are very actively researched in the database domain. Interested readers are referred to [1] for a detailed survey of spatiotemporal Databases.

Main Text

Contrary to traditional database systems, a spatiotemporal database must be able to manage the dynamic properties of spatiotemporal objects efficiently. As spatiotemporal objects constantly change, a spatiotemporal database usually receives intensive object updates in a short period of time. To reduce the number of object updates, a *sampling-based updating* method or a *velocity-based updating* method is usually adopted. The *sampling-based updating* method samples the spatial extent/location of a spatiotemporal object either periodically or whenever the spatial difference to the previous sample is greater than a pre-defined threshold. In the *velocity-based updating* method, a spatiotemporal object reports its location along with the velocity to a spatiotemporal database. The object does not issue a new report unless there is a velocity change. Besides the updating methods, spatiotemporal databases also employ novel access methods to efficiently store and update spatiotemporal objects [2].

Spatiotemporal databases support a wide range of novel queries over spatiotemporal data. Typical queries along the spatial dimension are Range Queries and K-Nearest-Neighbor (kNN) Queries. Typical queries along the temporal dimension are Historical Queries, Present Queries and Future Queries. Due to frequent updates of spatiotemporal data, queries in spatiotemporal databases often require continuous evaluation in the form of *continuous spatiotemporal queries*.

Compared to traditional database systems, spatiotemporal databases face special challenges in order to manage spatiotemporal data and to support corresponding queries efficiently. The challenges of spatiotemporal databases reside in and go beyond object representation, query languages, storage structure, indexing techniques, system models, query processing, query optimization and scalability issues [3].

Cross References

► Continuous Queries in Spatiotemporal Databases

Recommended Reading

1. Abraham, T., Roddick, J.F.: Survey of spatio-temporal databases. Geoinformatica 3(1): 61–99 (1999)
2. Mokbel, M.F., Ghanem, T.M., Aref, W.A.: Spatio-temporal access methods. IEEE Data Eng. Bull. **26**(2) (2003)
3. Sellis, T.K.: Research issues in spatio-temporal database systems. In: SSD '99: Proceedings of the 6th International Symposium on Advances in Spatial Databases, pp. 5–11. Springer, London (1999)

Spatio-temporal Database Modeling with an Extended Entity-Relationship Model

NECTARIA TRYFONA[1], CHRISTIAN S. JENSEN[2]
[1] Talent Information Systems SA, Athens, Greece
[2] Department of Computer Science, Aalborg University, Aalborg, Denmark

Synonyms

Conceptual modeling; Spatio-temporal modeling; Geographic databases

Definition

Applications dealing with space and time referenced data (e. g., location based services, cadastral systems, routing systems) benefit from conceptual modeling techniques that offer built-in support for the modeling of the spatial and temporal aspects of data. To support the spatial dimension, the new spatial concepts are the position, spatial properties, the fields and the spatial relationships, while for the temporal dimension, the temporal concepts of valid, transaction and existence time are needed. Applying these concepts in a meaningful way to the modeling constructs of objects, attributes, and relationships results in new modeling constructs, namely, spatial, temporal, and spatiotemporal objects, attributes, and relationships. The Spatio-temporal ER (STER) model extends the Entity-Relationship (ER) Model with the aforementioned spatiotemporal constructs.

Historical Background

The conceptual design phase focuses on expressing application requirements without the use of computer metaphors. The design should be understandable to the user and complete so that it can be translated into the logical phase that follows without any further user input. Popular conceptual modeling notations include the Entity-Relationship (ER) [2] and the Universal modeling Language (UML) [10].

For conventional administrative systems exemplified by the "supplier-supplies-parts" paradigm, the available modeling notations and techniques that support the conceptual and logical modeling phases are mature and adequate. However, this is not the case in non-standard systems managing spatiotemporal, multimedia, very-large-scale-integration (VLSI), image, and voice data. Rather, these lead to new and unmet requirements for modeling techniques.

The Chorochronos Project [5], established as a European Commission funded Training and Mobility of Researchers Network, had the objective of studying the design, implementation, and application of spatiotemporal database management systems. In this framework, Entity-Relationship was extended to accommodate the spatial, temporal and spatiotemporal dimension. A similar effort, extending UML towards spatiotemporal constructs, is also available [5,6].

Scientific Fundamentals

Spatial and Temporal Concepts

In the database realm, reality is a collection of objects characterized by a set of properties. Objects are interrelated via relationships.

Spatial Concepts Most real-world objects have a *position* or a *spatial extent*. In a spatiotemporal application, the positions of some objects *matter* and should be recorded, while the positions of other objects are not to be recorded. The former objects are termed *spatial* or *geographic objects* (*GO*).

The function p (position) takes spatial objects as arguments and returns the positions of the objects. Positions are parts of space and may be points, lines, regions, or combinations thereof, and are called *geometric figures*. Therefore, function p is defined as:

$$p : GO \rightarrow G ,$$

where G is the domain of geometric figures.

The embedding *space* must also be modeled in order to locate the objects in it. Space is modeled as a set and the elements of space are (*points*). Many different sets will do for space, but for practical reasons, space is modeled as a subset of R^2 or R^3 (i. e., two or three-dimensional Euclidean Space) in current spatial applications; in the present discussion, R^2 is used as space without affecting the generality of the approach.

An inherent property of spatial objects is that their positions may be viewed at different granularities and that the granularity affects the concrete data type of the position. For example, a "landparcel" may be seen as a point, a region, or both, depending on the granularity requirements of the application at hand. Such different *object views* have to be integrated into one conceptual description.

S

Spatial objects have *descriptive properties* and *spatial properties*. For example, in a cadastral application, a descriptive property is the "property category" or the "cadastral-id" of a landparcel, while a spatial property is the "soil type" of a landparcel. In the STER approach, spatial properties are properties of the embedding space that indirectly become properties of the spatial objects via their position in space, i. e., the spatial objects inherit them from space. For example, although one application may view the "soil type" of a landparcel as a property of the landparcel, it is clear that: (a) the "soil type" is defined whether or not the landparcel exists at that position in space, and (b) when the landparcel moves (or changes shape), the landparcel's "soil type" will not remain unchanged; rather, the "soil type" attribute inherits (or, obtains) new values from the new position.

The spatial properties of objects may be captured independently of the objects using so-called fields (the term *layer* is also used). Formally speaking, a field can be seen as a function from geometric figures to a domain of descriptive attribute values [1].

$$f_1 : G \rightarrow D_1 \times D_2 \times \cdots \times D_n \,.$$

In the function, G is the set of geometric figures and the D_i are (not necessarily distinct) domains. In other words, a field is a set of geometric figures with associated values. There are two basic types of fields:

a) those that are continuous functions, e. g., "temperature," or "erosion," and
b) those that are discrete functions, e. g., "county divisions" represented as regions.

In case (a), a field is a homogeneous (or continuous) area consisting of points, while in case (b), a field represents a set of regions with different values of the same attribute or positions of objects in space.

Finally, geographic objects may be related to each other in *space* via spatial (or *geographic*) *relationships*. For example, "the fjord Limfjorden *traverses* the city of Aalborg." Spatial relationships among geographic objects are actually relations on the objects' geographical extents.

The set of spatial relationships can be subdivided into three subsets: *topological* (e. g., "inside," "outside," etc.), *directional* (e. g., "North of," "North-East of," etc.), and *metric* (e. g., "5 km away from") relationships. Spatial relationships are further translated into spatial integrity constraints on the database.

Temporal Concepts Information about the properties of objects and the relationships among objects can be considered as *statements* or *facts* about the objects and their relationships.

For example, an application involving countries may include a "capital" property for the countries. The "Copenhagen" value of the property "capital" associated with "country" Denmark denotes the fact that "Copenhagen is the capital of Denmark." Precisely everything that can be assigned a truth value is a fact. For example, "Denmark is south of Greece" is a fact as it can be assigned a truth value (in this case, "false"). The sentence "Denmark and Greece are south" is not a fact.

The following three temporal aspects have been focused on within the research literature. They are universal and applications frequently require that these be captured in the database:

The *valid time* concept applies to facts: the valid time of a fact is the time when the fact was, is, or will be true in the modeled reality. For example, the valid time of "Copenhagen is the capital of Denmark" is (at least) the time from the year 1445 until the present day.

The *transaction time* concept applies not only to facts, but to any element that may be stored in a database: the transaction time of a database element is the collection of times when the element was or is a part of the current state of the database.

The *existence time* concept applies to objects: the existence time of an object is the times when the object existed or exists. Note that an object in itself cannot be true or false and therefore, cannot have a valid time. However, existence time and valid time are related: the existence time of an object O is the valid time of "O existed or exists." It should also be noted that some authors associate valid time with objects.

It is assumed that it only makes sense for an object to have properties and participate in relationships *when the object exists*. This implies that the valid times of facts associated with objects must be contained in the existence times for those objects.

Time values are drawn from a domain of time values, with the individual values being termed *chronons*. All three temporal aspects have duration and they may be captured using time intervals where a time interval $[t_s, t_e]$ is defined to be a set of consecutive chronons. Time values t_s and t_e are the start and the end chronons of the interval.

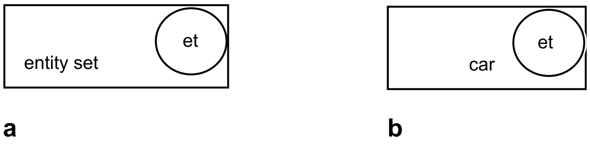
The following section extends ER in accordance with the foundations outlined above.

Spatiotemporal Entity-Relationship

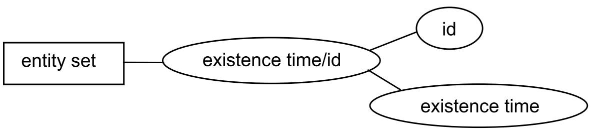
This section discusses the Spatiotemporal ER (STER), an extension of the Entity Relationship (ER) [2], to accommodate spatiotemporal peculiarities [5,7,8]. STER combines spatial and temporal concepts in a meaningful way. It also serves to explicitly state which aspects are indepen-

Spatio-temporal Database Modeling with an Extended Entity-Relationship Model, Table 1 Meaningful temporal aspects of ER constructs

	Entity set	Attribute	Relationship
Existence time	yes	no	no
Valid time	no	yes	yes
Transaction time	yes	yes	yes



Spatio-temporal Database Modeling with an Extended Entity-Relationship Model, Figure 1 **a** Capturing existance time and **b** representing car entities and their existance times



Spatio-temporal Database Modeling with an Extended Entity-Relationship Model, Figure 2 The ER diagram corresponding to Fig. 1a

dent and, by implication, which are not. It thus provides a guide for applying the general design criteria.

Enhancing ER with Spatiotemporal Constructs

The Spatiotemporal Entity Relationship (STER) [7,8,9,10] includes constructs with built-in spatial, temporal, and spatiotemporal functionality. A construct that captures a temporal aspect is called *temporal*; if it has built-in support for only a spatial aspect, it is termed *spatial*; and if it has both, it is *spatiotemporal*. The upper-right corner of each extended construct indicates its temporal support. The bottom-right corner indicates the spatial support.

While all basic constructs of ER can have spatial and temporal extents, not all temporal aspects are semantically meaningful for each construct. For example, existence time is applicable to precisely the entities in entity sets which are the only elements in ER with independent existence. Furthermore, an entity set may be given attributes that describe the properties of the set's entities. Earlier, it was stated that valid time is meaningful only for *facts*. When valid time is assigned to an attribute of an entity set, it indicates that the valid times of the facts are to be captured in the database. The same applies to attributes of relationship sets in place of entity sets. Finally, valid time may be assigned to a relationship set, indicating that the time when each relation in the set is true in the miniworld is to be captured in the database.

Transaction time applies to any "element" stored in the database, regardless of whether or not it may be assigned a *truth* value. Hence, unlike valid time, transaction time applies to entities in entity sets. Table 1 shows the meaningful combinations of temporal aspects and modeling constructs.

The abbreviations "et," "vt," "tt," and "bt" are used for "existence time," "valid time," "transaction time," and "bitemporal time," respectively. Abbreviation "bt" is shorthand for the combination of "vt" and "tt" that occurs often in a spatiotemporal database. Table 1 specifies the modeling constructs that each of these abbreviations applies to.

A. Entity Sets

Entity sets represent objects.

(i) Temporal Entity Sets

Entities in an entity set can be assigned existence and transaction time. The former is termed support for existence time, indicated by placing an "et" in a circle in the upper-right corner of the entity set's rectangle, as indicated in Fig. 1. Figure 1b shows that for "car" entities, existence time is to be captured.

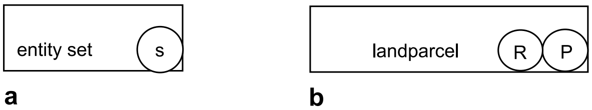
This notation is in effect shorthand for a larger ER diagram. This shorthand is convenient because it concisely states that the existence times of the entities in the entity set should be captured in the database. The more verbose ER diagram corresponding to the STER diagram in Fig. 1a is given in Fig. 2. Attributes connected to each other denote composite attributes 3.

Note that all the shorthand notations below can be transformed to the "full" ER diagrams. For reasons of easy reading, these verbose notations are omitted.

(ii) Spatial Entity Sets

Spatial objects have spatial extents and it is frequently necessary to capture these extents in the database. The first step to support this is to provide means for representing the space in which the objects are embedded. The next step is to provide means for indicating that the objects' extents in this space are to be captured. For these purposes, the following special entity and relationship sets are introduced.

1. The special entity sets are SPACE, GEOMETRY, POINT (or "P"), LINE (or "L"), and REGION (or "R"). Entity set GEOMETRY is used for capturing the shapes of entities and can be (i. e., is-a) POINT, LINE, REGION, or any other geometric type (or geometry). For simplicity, only POINT, LINE, REGION, and their combinations are used.
2. The special relationship set "is_located_at" associates a spatial entity set with its spatial extent or geometry. The cardinality of this set is 1:M, meaning that a spatial entity may have more than one geometry when multi-



Spatio-temporal Database Modeling with an Extended Entity-Relationship Model, Figure 3 **a** Spatial entity sets, **b** a landparcel as a POINT or a REGION

Spatio-temporal Database Modeling with an Extended Entity-Relationship Model, Figure 4 **a** A spatiotemporal entity set with valid-time support. **b** Recording car position with valid-time and transaction-time support

ple granularities are employed. Assuming that in each application there is only one space, the relationship set "belongs_to" between GEOMETRY and SPACE with cardinality M:1 is also included. When GEOMETRY is connected to SPACE it also captures the *locations* of objects. In this case, GEOMETRY describes the objects positions, i. e., shapes and locations.

The letters "s," "P," "L," or "R" in a circle in the lower-right corner of an entity set rectangle specify the spatial support. Letter "s" stands for SPATIAL and is used to indicate a spatial entity set whose exact geometric type is unknown. Letters "P," "L," "R," and their combinations specify geometric types as indicated above. These annotations may occur simultaneously and therefore represent different views of the same object. A spatial entity set is depicted in Fig. 3a. Figure 3b illustrates the spatial entity set "landparcel" with simultaneous geometries POINT and REGION; in this case, the representation in ER will only have REGION and POINT as geometries.

(iii) Spatiotemporal Entity Sets

For a geo-referenced object, GEOMETRY captures both its shape and location. When an object changes its position over time, i. e., the object moves (change of location) or the object changes shape, it is the GEOMETRY aspect of the object that changes rather than the object itself. For example, a car moving on a road changes its location, but this is not considered to change the car's identity. To capture a temporal aspect of the positions of the objects in an entity set, an "svt," an "stt," or an "sbt" is placed in a circle in the lower-right corner of the entity set's rectangle. The first annotation indicates valid-time support: the objects' current positions as well as their past and future positions are to be captured. This is illustrated in Fig. 4a. The second annotation (i. e., "stt") indicates transaction-time support: the current positions as well as all positions previously recorded as current in the database are to be captured. The third annotation (i. e., "sbt") indicates support for both valid and transaction time. Figure 4b shows that when a "car" changes positions the car's position in time is captured (i. e., "Pvt") as well as the time when this is recorded in the database (i. e., "Ptt"); this is indicated by "Pbt". If the geometric type of the entity set is known, the "s"-part is replaced by "P," "L," "R," or a combination thereof.

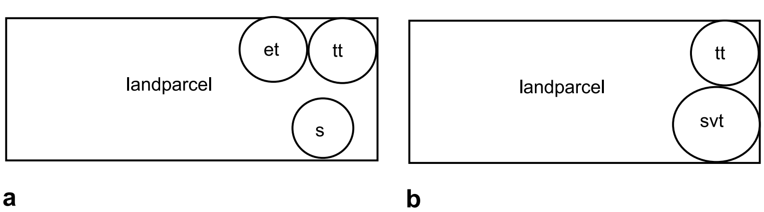
It is important to point out the difference between keeping track of (a) a spatial entity set in time, and (b) the position of a spatial entity set in time. In case (a) the temporal support refers to an entity's existence and recording in time. Figures 5a and b show the entity set "landparcel" as a spatial entity set (indicated by an "s"-part). Figure 5a illustrates that a cadastral database captures the existence time (i. e., when it first existed, indicated by "et") as well as the transaction time (i. e., when it was recorded in the database, indicated by "tt"). In contrast, Fig. 5b shows that for "landparcel," the database captures transaction time as before (indicated by "tt") as well as its geometry over time (indicated by "svt"). This means that if a landparcel changes shape in time, the (current) shape and the time this shape is true is recorded (valid time).

B. Attributes of Entity and Relationship Sets

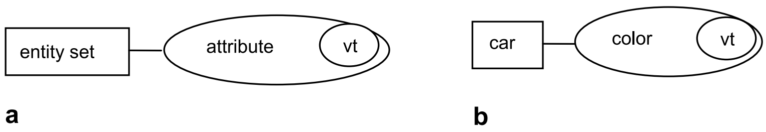
In a spatial environment, entity sets have two types of attributes: (a) *descriptive* attributes, such as the "cadastral-id" of a landparcel, and (b) *spatial* attributes, such as the "soil type" of a landparcel. The values of descriptive attributes for an entity (or a relationship) often change over time and it is often necessary to capture this in the database. Spatial attributes for which a temporal aspect is captured are termed *spatiotemporal attributes*.

(i) Temporal Descriptive Attributes

Values of attributes of entities denote facts about the entities and thus have both valid- and transaction-time aspects.



Spatio-temporal Database Modeling with an Extended Entity-Relationship Model, Figure 5 **a** A landparcel in space and time. **b** A landparcel in time, with position in time



Spatio-temporal Database Modeling with an Extended Entity-Relationship Model, Figure 6 **a** An attribute with valid-time support. **b** Car "color" with valid-time support

A circle with a "vt" or a "tt" in the upper-right corner of an oval denoting an attribute indicates the valid or transaction time, respectively, is to be captured. A circle with "bt" (bitemporal) indicates that both temporal aspects are to be captured. The sample STER diagram in Fig. 6a contains an attribute with valid-time support and Fig. 6b shows an example of keeping track of cars' colors and their valid time periods.

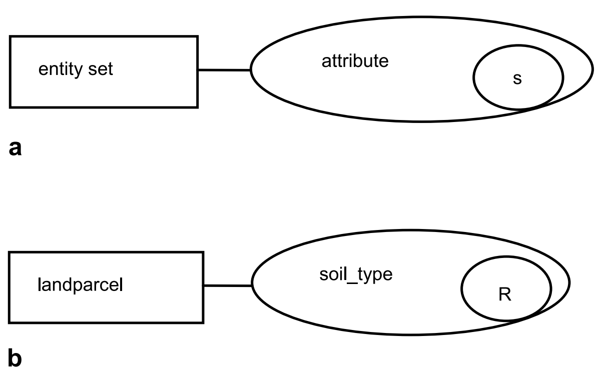
(ii) Spatial Attributes

Facts captured by attributes may also have associated locations in space which are described as sets of geometric figures. To capture this spatial aspect of an attribute, a circle with an "s" is used, as shown in Fig. 7a. Figure 7a. depicts the general representation of a spatial attribute, while Fig. 7b shows that the "soil type" value of a landparcel is associated with a set of spatial regions ("R").

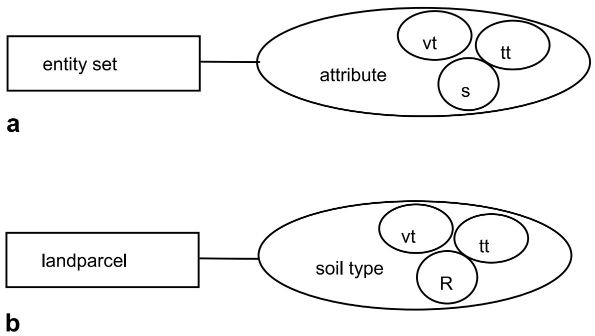
(iii) Spatiotemporal Attributes

Two cases are distinguished here. The first case concerns spatial attributes with temporal support that refers to the attributes' valid- and transaction-time periods (i. e., the spatial attribute is treated as a normal attribute in time). This is illustrated in Fig. 8a. Figure 8b provides an example.

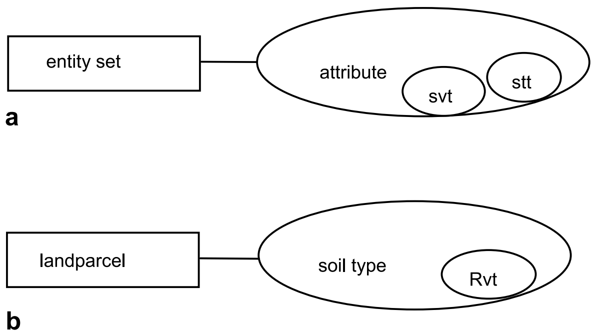
In the second case, the temporal aspects (valid and transaction time) of spatial attributes are recorded by placing "svt," "stt," or "sbt" (and replacing the "s" with "P," "L," or "R," or a combination of these if the geometric types of the geometric figures of the attributes are known) in same way as for entity sets. This is illustrated in Fig. 9a and Fig. 9b provides an example.



Spatio-temporal Database Modeling with an Extended Entity-Relationship Model, Figure 7 **a** A spatial attribute in STER. **b** "soil type" as a spatial attribute



Spatio-temporal Database Modeling with an Extended Entity-Relationship Model, Figure 8 **a** A spatial attribute with temporal support. **b** "soil type" with temporal support



Spatio-temporal Database Modeling with an Extended Entity-Relationship Model, Figure 9 **a** A spatial attribute with temporal support for its spatial part. **b** "soil type" as a spatiotemporal attribute

C. Relationship Sets

(i) Temporal Relationship Sets

By annotating a relationship set with a temporal aspect (valid time, transaction time, or both), the changes of the set's relationships with respect to that aspect are captured.

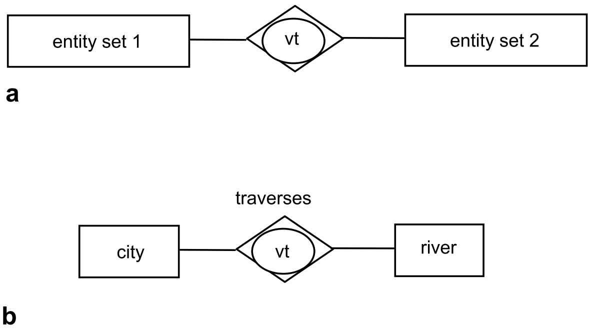
(ii) Spatial Relationship Sets

Spatial relationship sets are special kinds of relationship sets. In particular, they are associations among the geometries of the spatial entities they relate. For reasons of simplicity and ease of understanding, spatial relationship sets are given as relationships among the spatial entity sets themselves. For example, the relationship "traverses" between cities and rivers relates the geometries of entities of these two spatial entity sets.

S

(iii) Spatiotemporal Relationship Sets

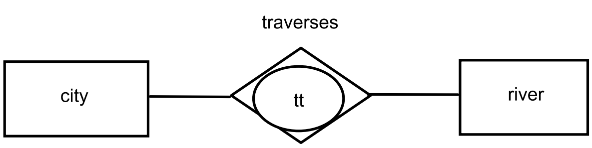
A spatiotemporal relationship set is a spatial relationship set with time support. In particular, by annotating a spatial relationship set with a temporal aspect, the changes of the spatial relationship with respect to that aspect are captured. Figure 10a shows the general representation of a spatiotemporal relationship set, while Fig. 10b depicts changes of the relationship "traverses" between cities and rivers recorded in time.



Spatio-temporal Database Modeling with an Extended Entity-Relationship Model, Figure 10 **a** A spatiotemporal relationship set in STER. **b** "traverses" as a spatitemporal relationship set

Finally, the previous discussion regarding temporal, spatial, and spatiotemporal attributes applies also to attributes of relationship sets.

The description of STER thus far has primarily focused on adding existence and valid time to ER constructs. The temporal aspect of transaction time is not strictly part of capturing the modeled reality. However, capturing transaction time, as reflected in the systems requirements, is important when designing a real-world information system and an ER should also provide built-in support for the capture of this aspect. Transaction time is applicable in STER exactly where valid time or existence time is applicable. For example, to capture the transaction time of a relationship set, in Fig. 10a, all that is needed is the addition of "tt" to the relationship set construct (Fig. 11).



Spatio-temporal Database Modeling with an Extended Entity-Relationship Model, Figure 11 "Traverses" as a spatiotemporal relationship set with transaction time

A Textual Notation for STER

In order to fully support STER, its graphical notation as well as its textual notation has to be provided. In this way, all the constructs of the diagrammatic STER can be presented and/or transformed into the textual notation. The full explanation is given in [9].

The definitions use the following meta-syntax. Upper case words are reserved words and lower case words are variables that represent arbitrary names. Capitalized words in lower case denote variables with restricted range. Optional elements are enclosed in < > and (, . . .) denotes repeatable elements; the notation {. . . | . . . } denotes selection (one of), and ( ) indicates the grouping of arguments.

```
DEFINE ENTITY SET entity_set_name
<TYPE Entity_construct (entity_set_name_i, . . . )>
<ATTRIBUTES
  ((attribute_name_j
    <VALID_TIME> <TRANSACTION_TIME>
    <GEOMETRY Geometric_type
      <VALID_TIME><TRANSACTION_TIME>>), . . . )>
<GEOMETRY Geometric_type
      <VALID_TIME><TRANSACTION_TIME>>
<EXISTENCE_TIME><TRANSACTION_TIME>
<AS ISA OF (entity_set_name_k, . . . )>>

DEFINE ATTRIBUTE attribute_name_m
  <AGGREGATION_OF ((attribute_name_n <VALID_TIME
                                          Valid_time>
    <TRANSACTION_TIME Transaction_time>
    <GEOMETRY Geometric_type
    <VALID_TIME><TRANSACTION_TIME>>), . . . )>

DEFINE <SPATIAL> RELATIONSHIP SET
  relationship_set_name (entity_set_name_i, . . . )
  TYPE Relationship_type
  <ATTRIBUTES ((attribute_name_k
    <VALID_TIME><TRANSACTION_TIME>), . . . )>
  <VALID_TIME><TRANSACTION_TIME>
```

Key Applications

Modeling of spatial and temporal concepts is essential for every application dealing with space. Some characteristic applications are cadastral applications, fleet management systems, crisis management systems, utility networks applications, and, recently, mobile applications, e. g., location based services for tourism.

Future Directions

Location-based services and fleet management are new, emerging applications calling for spatiotemporal support. The modeling needs of these applications may pose new challenges to conceptual modeling notations. As an example, current locations of mobile objects are sampled according to some specific approach. How to capture the accuracy of the data and the sampling approach employed in a conceptual model is a new challenge. Accommodating and building constructs per-customer's needs is the next step in this direction.

Cross References

► Cadastre
► Modeling with ISO 191xx Standards
► Temporal GIS and Applications

Recommended Reading

1. Delis, V., Hadzilacos, T.H., Tryfona, N.: An Introduction to Layer Algebra. In: Proceeding of the 6th International Conference on Spatial Data Handling (SDH). Edinburgh, Scotland 5”9 Sept 1995
2. Chen, P.S.: The Entity-Relationship Model: Toward a unified view of Data., ACM Trans. Database Sys. **1**, 9–36 (1976)
3. Elmasri, R., Navathe, S.: Fundamental of Database Systems (5. edn). Addison Wesley, Boston, MA, USA (2007)
4. Hadzilacos. T.H., Tryfona, N.: Extending the Entity-Relationship Model to Capture Spatial Semantics., SIGMOD RECORD. 26 (3), 24”29 (1997)
5. Koubarakis, M., Sellis, T., Frank, A.U., Grumbach, S., Güting, R.H., Jensen, C.S., Lorentzos, N.A., Manolopoulos, Y., Nardelli, E., Pernici, B., Schek, H.J., Scholl, M., Theodoulidis, B., Tryfona, N. (eds.): Spatio-temporal Databases: The CHOROCHRONOS Approach. Lecture Notes in Computer Science 2520, Springer, Berlin (2003) ISBN 3-540-40552-6
6. Price, R., Tryfona, N., Jensen, C.S.: Extended Spatiotemporal UML: Motivations, Requirements, and Constructs. J. Database Manage. **11** (4), 14–17 (2000)
7. Tryfona, N., Jensen, C.S.: Using Abstractions for Spatio-temporal Conceptual Modeling. In: Proceedings of the 14th Annual Symposium on Applied Computing (SAC'00). Villa Olmo, Como, Italy (2000)
8. Tryfona, N., Jensen, C.S.: Conceptual Data Modeling for Spatiotemporal Applications. Geoinformatica **3** (3). (1999)
9. Tryfona, N., Price, R., Jensen, C.S.: Spatiotemporal Conceptual Modeling.In: Spatiotemporal Databases: The CHOROCHRONOS approach. Springer-Verlag, Berlin (2003)
10. Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language, Reference Manual, 2nd edn. Addison Wesley, Boston, MA, USA (2005)

Spatio-temporal Databases

CINDY X. CHEN
Department of Computer Science,
University of Massachusetts, Lowell, MA, USA

Definition

A spatio-temporal database is a database that embodies spatial, temporal and spatio-temporal database concepts and captures simultaneously spatial and temporal aspects of data. It deals with geometries changing over time.

Spatio-temporal query languages are the query languages used to query spatio-temporal databases. For example, “when did the fire called “The Big Fire” reach its largest extent?” is a typical spatio-temporal query.

Spatio-temporal query languages are usually implemented as extensions to the standard relational query language SQL.

Main Text

In conventional databases, attributes containing temporal or spatial information are solely manipulated by the application programs. A spatio-temporal database is one that supports aspects of both time and space. It offers spatial and temporal data types in its data model and query language.

Temporal data types include time instants and time periods which represent a continues period of time; spatial data types include polyhedrons, polygons, lines and points. Other spatial shapes can be approximated by polyhedrons or polygons. Spatio-temporal data types include moving points and moving regions.

Spatio-temporal query languages support spatial operators such as CONTAIN, OVERLAP, ADJACENT, DISTANCE, CIRCUMSTANCE, AREA, VOLUME, etc.; temporal operators such as CONTAIN, OVERLAP, MEET, BEFORE, AFTER, DURATION, etc.; and spatio-temporal operators such as MOVING_DISTANCE, etc. These operators are either supported as user-defined functions or as user-defined aggregates of a conventional relational database or as a database extender.

Cross References

► Spatio-temporal Query Languages

Spatio-temporal Index

► Mobile Object Indexing

Spatio-temporal Index Structures

► Indexing Spatio-temporal Archives

Spatio-temporal Indexing

► Indexing and Mining Time Series Data
► Indexing, Query and Velocity-Constrained
► Indexing the Positions of Continuously Moving Objects

Spatio-temporal Informatics

► Temporal GIS and Applications

Spatio-temporal Information Systems

► Temporal GIS and Applications

Spatio-temporal Interpolation

▶ Constraint Databases and Data Interpolation

Spatio-temporal Modeling

▶ Spatio-temporal Database Modeling with an Extended Entity-Relationship Model

Spatio-Temporal Modeling Language Extension

▶ Modeling with Pictogrammic Languages

Spatio-temporal Network Coding

▶ Spatio-temporal Queries on Road Networks, Coding Based Methods

Spatio-temporal Object Association

▶ Patterns in Spatio-temporal Data

Spatio-temporal Objects

▶ Constraint Databases and Moving Objects

Spatio-temporal Queries on Road Networks, Coding Based Methods

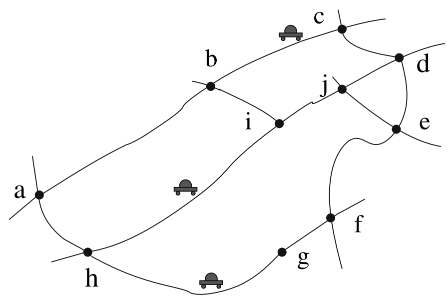
SANDEEP GUPTA, CHINYA V. RAVISHANKAR
University of California, Riverside, Riverside, CA, USA

Synonyms

Spatial and spatio-temporal queries over moving objects; Spatio-temporal network coding

Definition

Coding-based techniques assign codes to intersections on road networks, so that shortest distance between intersections can be determined very quickly as Hamming distances [9] between their codes. Conventional methods have been used when distances between objects are defined under the Euclidean metric, but coding-based methods are superior for join, range, intercept, and other spatial and spatiotemporal queries when distances are measured *along the roads*.



Spatio-temporal Queries on Road Networks, Coding Based Methods, Figure 1 A simple road network

Node	Code
a	01011111111011
b	00000001011011
c	00000000000111
d	00000000000100
e	10000000000000
f	11111000000000
g	11111110000000
h	01011111111000
i	00000001011000
j	00000000001000

Spatio-temporal Queries on Road Networks, Coding Based Methods, Figure 2 The road network embedded into H_{14}

The coding is obtained by embedding the graph metric into a higher-dimensional space, such as a hypercube of a suitable dimension. As an example, Fig. 2 depicts the labels for nodes from the sample road network in Fig. 1, obtained using an isometric embedding of the road network into a hypercube of dimension 14.

Typical spatiotemporal range queries have the form *"Find all objects that are in region R during a time interval* $[t_1, t_2]$", "*Find all objects that are within distance d of object* o_1", or *"Find all objects that are encountered by object* o_1 *in the time interval* $[t_1, t_2]$". Similarly, join queries have the form *"Find all pairs of objects which pass a common waypoint"*, or *"Find all pairs of objects that are within distance d of one another at a given time instant t"*, and typical intercept queries have the form *"Find the time that it will take for object* o_1 *to catch* o_2".

Historical Background

Research on queries on moving objects has included data models and query languages [2,4,5,12,15] index structures, and efficient query processing techniques [10,13,14,19]. Traditional work in indexing moving objects [1,14] tends to use simple motion models, which are often linear. However, index structures for moving data are complex because they must deal with continuously changing positions, and an additional *time* dimension. Shortest-distance computation is critical for spatiotemporal query processing when distances are computed along the road network. Some work [11,14,16], which assumes unrestricted motion, can be useful as a first-cut filter.

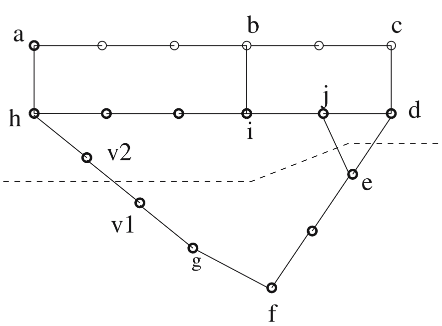
Research on graph algorithms has addressed the problem of accelerating shortest distance computation over graphs, especially for planar graphs. Dijkstra's celebrated algorithm [3], requires no preprocessing and takes $O(n)$ space and $O(n \log n)$ time, when implemented with heaps. An $O(n\sqrt{\log n})$ time algorithm [6] that works for planar graphs is the most practical and efficient of all the algorithms available. However, since we would like to be able to handle very large networks, we would prefer a response time even smaller than provided by current methods. One simple approach might be to compute shortest paths in advance and do a simple table lookup to give a constant time response. However, this method is also impractical since even a 100,000 node network would require a table with five billion entries.

Scientific Fundamentals

We illustrate the basics of this method using an example. For a comprehensive discussion of this material, see [8].

A Coding Example

We first represent the road network in Fig. 1 as a planar graph (see Fig. 3). Distances in the planar graph are assigned by discretized distances in the road network, keeping in mind the precision requirements. Codes are generated using a series of *alternating cuts* illustrated in Fig. 3, and explained in detail in [8]. The value of each bit position in the code corresponding to a graph node is determined by the disposition of the node with respect to an alternating cut. All nodes on one side of the alternating cut are arbitrarily assigned the value 0 for that bit position, and all nodes on the other side of the cut are assigned the value 1. The length of the code is hence equal to the number of alternating cuts, which is determined by the number of edges on the outer face of the graph.



Spatio-temporal Queries on Road Networks, Coding Based Methods, Figure 3 Converting a road map into its corresponding planar graph

Figure 2 lists the codes for all nodes. The Hamming distance between any two codes is twice the shortest distance between the corresponding vertices of the graph. For example, the shortest distance on the graph between nodes a and f is 5. The corresponding codes for a and f are 01011111111011 and 11111000000000, which have a Hamming separation of 10. The validity of this encoding for other pairs of points may be verified.

Processing Spatiotemporal Queries

Next, we overview how codings can be applied to process spatiotemporal queries. Let graph vertex v_i be assigned code $c(v_i)$, and let the Hamming distance between codes $c(v_i)$ and $c(v_j)$, be represented as $|(c(v_i), c(v_j)|$. For each vertex v_i we maintain a set S_i of its adjacent nodes.

Since objects move only along graph edges, their positions can be defined by distances along edges. A position p_i on edge $e_j = (v_{j_1}, v_{j_2})$ is assigned coordinates of the form (v_{j_1}, v_{j_2}, d_i) where d_i indicates how far along e_j the position p_j is located. The shortest distance between two points can now be trivially found with just four computations of shortest distance between vertices.

Trajectories in road networks [17] are generally represented as collections of line segments. This approach incurs overhead, and significantly lowers overall performance. Coding allows a simpler approach.

Let $\langle i, j\rangle$ represent the shortest path between vertices v_i and v_j. Routes can be defined to be shortest paths between the source and destination vertices, so $\langle i, j\rangle$ would be the default route between v_i and v_j. To accommodate routes that differ from this shortest path, one may define *waypoints*, and require objects to follow the shortest path between waypoints. The route for an object that moves between vertices v_{i_1} and v_{i_m}, via waypoints $v_{i_2}, \ldots, v_{i_{m-1}}$, is represented as $\langle i_1, i_2, \ldots, i_{m-1}, i_m\rangle$, and is obtained by concatenating the shortest paths $\langle i_1, i_2\rangle, \langle i_2, i_3\rangle, \ldots, \langle i_{m-1}, i_m\rangle$.

For reasonable vehicular paths, the number of waypoints is likely to be low. Further, if several shortest paths exist between the given (source, destination) pair, one can force a choice between them by using waypoints.

As we will see below, query processing depends on the following observation. A vertex v lies on a shortest path $\langle i, j\rangle$ if and only if $|c(v_i, v_j)| = |c(v_i, v)| + |c(v, v_j)|$. In this case, we say $\langle i, j\rangle$ *includes* v. Similarly, $\langle i_1, i_2, \ldots, i_m\rangle$ includes v if and only if there are two consecutive way points $v_{i_k}, v_{i_{k+1}}$ such that $\langle k, k+1\rangle$ includes v.

The Route Hashing Optimization

As noted above, bit position k of a vertex's code is determined by the k^{th} alternating cut, which divides the road network into two components. All vertices in one component will contain a 0 in bit position k of their codes, and all vertices in the other component will contain a 1. We can immediately determine, from bit position k of its code, which component a vertex falls into.

We next observe that for a class of well-behaved graphs (having the *isometric* property), the shortest path between two vertices in the same component will not pass through any node in the other component. We can use this observation to prune the search space of a query, using a framework reminiscent of hashing. The idea is to create a set of "bins" and a hash function to maps each shortest path to a subset of these bins. This idea is developed fully in [8].

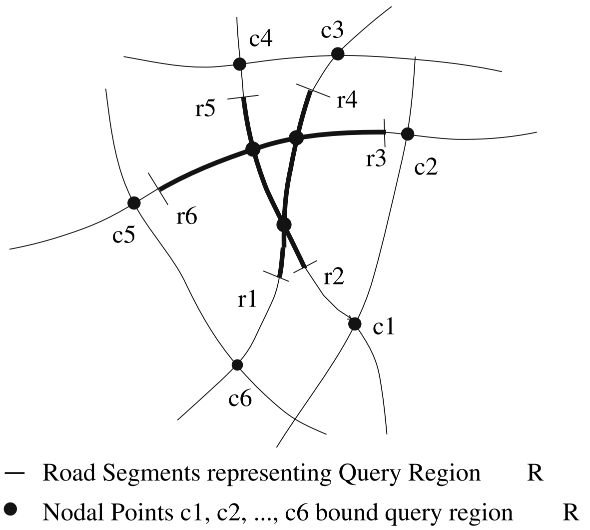
As a simple illustration of this idea, let an object start from vertex v_1 and travel to vertex v_2. Let the codes from v_1 and v_2 both contain the same value $b \in \{0, 1\}$ in bit position k. If the code for a vertex v_3 contains $\bar{b}$ in bit position k, we know immediately that the object will not pass through v_3.

Query Processing

We will illustrate query processing using incidence queries. Given a query vertex v_q and time interval $[t_1, t_2]$, an incidence query requests all objects that pass through v_q during $[t_1, t_2]$. To answer the incidence query, we first pick the bins whose range contains v_q (this can be done by using the codes, as in [8]). For each route $\langle o_i \rangle$ in these bins, we first check if $\langle o_i \rangle$ includes v_q, and second, whether o_i could have passed v_q during $[t_1, t_2]$.

Let us denote the route taken by object o_i as $\langle o_i \rangle$, and the distance of vertex v along the route $\langle o_i \rangle$ by $dist(\langle o_i \rangle, v)$. If object o_i travels at speed $u_i(t)$, and passes through vertex v at time t_v, we will have $dist(\langle o_i \rangle, v) = \int_0^{t_v} u_i(t)\,dt$. In answering incidence queries, we would check whether $\int_0^{t_1} u_i(t)\,dt \leq dist(\langle o_i \rangle, v_q) \leq \int_0^{t_2} u_i(t)\,dt$.

We can also address range, join and intercept queries using the framework in [8].



Spatio-temporal Queries on Road Networks, Coding Based Methods, Figure 4 Highlighted segment represents query region

Key Applications

The primary application of this technology are location-based services such as fleet management, travel guides, and navigation systems. Such applications typically monitor the positions of objects of interest on road networks, such as cars, cargo, or travelers, using positioning systems and wireless communications. Such systems often issue location-dependent queries, asking for the locations of the nearest site of interest (in terms of distance or time), such as a warehouse, gas station, or restaurant. In mobile ad-hoc networks, the sites of interest may also be mobile. In this case, query processing will require very rapid computation of future positions of both the objects as well as the sites of interest.

Scalable solutions for location-dependent information delivery to mobile objects will be a problem of increasing importance in the future. The importance of data modeling, trajectory representation and query operators for future applications are well-recognized [18]. Coding techniques are very useful for trajectory representation and some of the query operators discussed in [18] when objects are moving along the road network.

Future Directions

Moving object databases will play a central role in enabling scalable location-based services, which are poised to become pervasive, given continued growth technologies such as GPS and wireless handheld devices. Possibilities for future work include generalizations of these techniques and those in [7] to handle more complex classes of graphs, and ultimately, to real road networks with dynamic edge weights. The method also needs to accommodate and model imprecision in inputs such as locations or values, and provide query responses with accompanying error estimates and confidence intervals.

Cross References

► Maximum Update Interval in Moving Objects Databases
► Spatio-temporal Query Languages
► Trip Planning Queries in Road Network Databases

Recommended Reading

1. Agarwal, P.K., Arge, L., Erickson, J.: Indexing moving points. In: Proc. of the 19th ACM Symp. on Principles of Database Systems (PODS), pp. 175–186 (2000)
2. Chen, C.X., Zaniolo, C.: SQL ST: A spatio-temporal data model and query language. In: International Conference on Conceptual Modeling / the Entity Relationship Approach, pp. 96–111 (2000)
3. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to Algorithms. MIT Press, McGraw-Hill, Cambridge, London (1990)
4. Erwig, M., Guting, R.H., Schneider, M., Vazirgiannis, M.: Spatio-temporal data types: An approach to modeling and querying moving objects in databases. Geoinformatica **3**(3), 269–296 (1999)
5. Forlizzi, L., Guting, R., Nardelli, E., Schneider, M.: A data model and data structures for moving objects databases. In: Proc. of ACM SIGMOD, pp. 319–330 (2000)
6. Frederickson, G.N.: Fast algorithms for shortest paths in planar graphs, with applications. In: SIAM J. Computing 16(6), pp. 1004–1022 (1987)
7. Gavoille, C., Peleg, D., Perennes, S., Raz, R.: Distance labeling in graphs. In: Symposium on Discrete Algorithms, pp. 210–219 (2001)
8. Gupta, S., Kopparty, S., Ravishankar, C.: Roads, codes, and spatiotemporal queries. In: PODS '04: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 115–124, ACM Press, New York, NY, USA (2004)
9. Hamming, R.W.: Error-detecting and error-correcting codes. In: Bell System Technical Journal, pp. 29(2):147–160. Bell System (1950)
10. Jensen, C.S., Lin, D., Ooi, B.C.: Query and update efficient b+-tree based indexing of moving objects. In: vldb, pp. 768–779 (2004)
11. Kollios, G., Gunopulos, D., Tsotras, V.: On Indexing Mobile Objects. In: Proc. of the 18th ACM Symp. on Principles of Database Systems (PODS), pp. 261–272, June (1999)
12. Christine, P., Spaccapietra, S., Zimnyi, E.: Spatio-temporal conceptual models: data structures + space + time. In: GIS '99: Proceedings of the 7th ACM international symposium on Advances in geographic information systems, pp. 26–33, ACM Press, New York, NY, USA (1999)
13. Prabhakar, S., Xia, Y., Kalashnikov, D., Aref, W., Hambrusch, S.: Query indexing and velocity constrained indexing: Scalable techniques for continuous queries on moving objects. IEEE Transactions on Computers **51**(10):1124–1140, October (2002)
14. Saltenis, S., Jensen, C., Leutenegger, S., Lopez, M.A.: Indexing the Positions of Continuously Moving Objects. In: Proceedings of the ACM SIGMOD, pp. 331–342, May (2000)
15. Sistla, A., Wolfton,O., Chamberlain, S., Dao, S.: Modeling and querying moving objects. In: Proceedings of the 13th International Conference on Data Engineering (ICDE'97), pp. 422–433. IEEE, Washington, Brussels, Tokyo, April (1997)
16. Tao, Y., Papadias, D., Sun, J.: The tpr*-tree: An optimized spatio-temporal access method for predictive queries. In: Proc. of the VLDB (2003)
17. US Census Bureau. TIGER. http://tiger.census.gov/
18. Wolfson, O.: Moving objects information management: The database challenge. In: NGITS '02: Proceedings of the 5th International Workshop on Next Generation Information Technologies and Systems, pp. 75–89. Springer, London (2002)
19. Zhou, P., Zhang, D., Salzberg, B., Cooperman, G., Kollios, G.: Close pair queries in moving object databases. In: GIS '05: Proceedings of the 13th annual ACM international workshop on Geographic information systems, pp. 2–11. ACM Press, New York, NY, USA (2005)

Spatio-temporal Query

► Queries in Spatio-temporal Databases, Time Parameterized

Spatio-temporal Query Languages

CINDY X. CHEN
Department of Computer Science,
University of Massachusetts, Lowell, MA, USA

Synonyms

Spatio-temporal data models; SQL/MM; SQL/Temporal; Operator, spatial; Operator, temporal; Bi-temporal; ISO; Operator, topological; Operator, metric

Definition

A spatio-temporal database is a database that embodies spatial, temporal and spatio-temporal database concepts and captures simultaneously spatial and temporal aspects of data. Spatio-temporal databases allow sophisticated queries over time and space to be stated [6]. Spatio-temporal query languages are the query languages used to query spatio-temporal databases.

Though we could use the basic data types in SQL to represent spatio-temporal data, however, the queries would be very complicated to write. In some cases, we may not be able to answer some queries. For example, assume there is a traveler. If we use date to represent the start and end of a time interval while the person remains at the same place, and two floating point numbers to represent the coordinates of a point in a two-dimensional space which is the position of the person, the table will have two more columns than if we use a temporal data type interval and a spatial data type point. The query "find if two travelers have ever crossed path" will have four more lines in the former case than in the latter case . Assume that there is a city which has a diamond shape. If we use eight floating point numbers to represent the vertexes of the diamond, the query "find if a traveler has ever been in Diamond City" will be very difficult to write. On the other hand, if we use a polygon to represent the city, we will be able to use the spatial operator "contain" to test whether the traveler is inside the city.

S

Spatio-temporal databases support spatial and temporal data types in addition to traditional data types. Temporal data types include time instants and time intervals or periods which are a continuous period of time. Spatial data types include points, lines which are finite line segments and polygons. Other spatial objects can be approximated by polygons.

Spatio-temporal query languages support a set of operators. Temporal operators [1] include CONTAIN, OVERLAP, MEET, BEFORE, AFTER, DURATION, etc. Spatial operators [10] include CONTAIN, OVERLAP, TOUCH/ADJACENT, DISJOINT, DISTANCE, etc. Spatio-temporal operators include MOVING_DISTANCE, etc.

For example, the following are two sample spatio-temporal queries:

Query 1: Determine the times and locations when "The Big Fire" started.

```
SELECT time, extent
FROM forest_fire
WHERE firename="The Big Fire"
HAVING time=(SELECT MIN(time)
  FROM forest_fire
  WHERE firename="The Big Fire")
```

Query 2: How long was fire fighter Mr. Miller enclosed by the fire called "The Big Fire" and which distance did he cover there?

```
SELECT DURATION(f1.time),
  MOVING_DISTANCE(f2.location, f2.time)
FROM forest_fire AS f1, fire_fighter AS f2
WHERE f1.time = f2.time
AND f1.firename = "The Big Fire" AND
  f2.fightername = "Mr. Miller"
GROUP BY f1.time
HAVING CONTAIN(f1.extent, f2.location)
```

Spatio-temporal query languages usually are implemented as extensions to the standard relational query language SQL.

Historical Background

In the past 20 years, many data models and corresponding query languages for spatio-temporal databases have been proposed. Among them, there are three major approaches – constraint-based approach, composite spatio-temporal data types, and orthogonal time and space.

Constraint databases can be used as a common language layer that makes the interoperability of different temporal, spatial and spatio-temporal databases possible. Constraint databases generalize the classical relational model of data by introducing generalized tuples: quantifier-free formulas in an appropriate constraint theory. [3] first introduced the TQuel data model for temporal databases and the 2-spaghetti model [12] for spatial databases, and then developed a parametric 2-spaghetti data model for spatio-temporal data. The approach presented in [3] was extended in [2] by using n-dimensional parametric rectangles (or boxes). A n-dimensional rectangle is the cross product of n intervals, each in a different dimension. The lower and upper bounds of the intervals are functions of time.

In [7,8,9], a framework of abstract data types for moving objects was defined. The framework takes as its outset a set of basic types that along with standard data types, such as integer and boolean; includes spatial data types, such as points and regions, and temporal data types, such as time instants. Then, it introduces type constructors that can be applied to the basic types to create new types. For example, given an argument of type α, the function "moving" constructs a type whose values are functions of time into the domain of α. This leads to types such as *mpoint* (moving point) and *mregion* (moving region). These abstract types may be used as column types in conventional relational DBMSs or they may be integrated into Object-Oriented or Object-Relational DBMSs.

In [14], a unified model is presented for information that uses two spatial dimensions and two temporal dimensions (valid time and transaction time). The temporal objects in this 4-dimensional space are called *bitemporal elements*. A *simplex* is either a single point, a finite straight line segment, or a triangular area. A simplicial complex is a collection of non-overlapping simplexes. A spatio-temporal object is a unified object which has both spatial and bitemporal extents. An *ST-simplex* is an elemental spatial object with a bitemporal reference attached. An *ST-complex* is a collection of ST-simplexes subject to some constraints.

In [4,5], the model proposed by Worboys was extended. Each state of a spatial object is captured as a snapshot of time; then, a directed-triangulation model is used to represent spatial data and a point-based model is used to represent time at the conceptual level. Spatio-temporal reasoning and queries are fully expressed with no new constructs, but user-defined aggregates, such as AREA and CONTAIN for spatial relationships, DURATION and CONTAIN for temporal ones, and MOVING_DISTANCE for spatio-temporal ones queries.

Unlike SQL/MM Spatial [13] and SQL/Temporal, there has not been much effort towards standardizing spatio-temporal support in SQL. However, the ISO standard "Geographic information – Temporal schema" defines concepts for describing temporal characteristics of geographic information.

Scientific Fundamentals

Table 1 shows the semantics of temporal operators.

Spatio-temporal Query Languages, Table 1 Semantics of Allen's operators on PERIOD

Operator	*Semantics defined on* PERIOD
DURATION(*x*)	DURATION: PERIOD $\longmapsto$ INT *end(x) - start(x)*
x OVERLAP *y*	OVERLAP: PERIOD × PERIOD $\longmapsto$ BOOLEAN $start(x) < end(y) \wedge start(y) < end(x)$
x PRECEDE *y*	PRECEDE: PERIOD × PERIOD $\longmapsto$ BOOLEAN $start(x) < start(y)$
x CONTAIN *y*	CONTAIN: PERIOD × PERIOD $\longmapsto$ BOOLEAN $start(x) \leq start(y) \wedge end(x) \geq end(y)$
x MEET *y*	MEET: PERIOD × PERIOD $\longmapsto$ BOOLEAN $start(x) = end(y) \vee end(x) = start(y)$
x EQUAL *y*	EQUAL: PERIOD × PERIOD $\longmapsto$ BOOLEAN $start(x) = start(y) \wedge end(x) = end(y)$
x INTERSECT *y*	INTERSECT: PERIOD × PERIOD $\longmapsto$ PERIOD *x* OVERLAP *y* $\models (max(start(x), start(y)), min(end(x), end(y)))$

Spatial operators can be divided into two types – topological ones, which define relationships between two spatial objects and metric ones, which measure the property of a spatial object. Topological spatial operators are defined in a similar way as the temporal operators:

- `equal(object1, object2)` — iff the set of the vertexes and set of edges of object1 are equal to those of object2
- `overlap(object1, object2)` — iff at least one vertex of object2 is *inside* object1 or at least one edge of object2 *crosses* with an edge of object1
- `contain(object1, object2)` — iff all the vertexes of object2 are *inside* object1
- `disjoint(object1, object2)` — iff non of the vertexes of object1 is *inside* object2 and vice versa; and non of the edges of object1 *crosses* with any edge of object2
- `adjacent(object1, object2)` — iff one edge of object1 *overlaps* with an edge of object2 and at least one vertex of object1 is not *inside* object2
- `commonborder(object1, object2)` — iff an edge of object1 is *equal* to an edge of object2
- `meet(object1, object2)` — iff one vertex of object1 is *on* an edge of object2

To determine if a point is *inside* a polygon or two lines *cross*, methods developed in computational geometry are used. Primitives such as *inside* and *cross* and metric operators such as distance, perimeter, area, etc., are implemented as external functions using functional programming language.

Key Applications

There are three major types of applications for spatio-temporal data models and query languages.

The first one is applications that involve moving objects such as navigational systems and objects that change position but not shape, for example, sensor network or wireless phone/computer users.

The second type of applications are those dealing with discrete changes of an object's shape and position, for example, countries' boundaries may change over throughout history.

The last type of applications are those integrating continuous motion as well as changes of shape, for instance, monitoring a forest fire or a tropical storm.

Future Directions

There is not much research done on three-dimensional spatio-temporal databases, though there are a lot of applications in atmospheric sciences, geology, meteorology, medical imaging, and space physics, etc. Because of the "*curse of dimensionality*", increased dimensionality in space poses new challenges for designing a data model and query language. The data model should be defined so that three-dimensional spatio-temporal data can be efficiently stored and effectively queried. The query language should be easy to implement as an extender to standard SQL.

Cross References

► Queries in Spatio-temporal Databases, Time Parameterized

Recommended Reading

1. Allen, J.F.: Maintaining knowledge about temporal intervals. In: Communications of the ACM, **26**(11), pp. 832–843. ACM Press, New York, NY, USA (1983)
2. Cai, M., Keshwani, D., Revesz, P.Z.: Parametric Rectangles: A Model for Querying and Animation of Spatiotemporal Databases. In: Proceedings of the 7th International Conference on Extending Database Technology, pp. 430–444. Konstanz, Germany (2000)
3. Chomicki, J., Revesz, P.Z.: Constraint-Based Interoperability of Spatiotemporal Databases. In: Scholl, M., Voisard, A. (eds.) Advances in Spatial Databases, pp. 142–161. Springer-Verlag, Berlin, Heidelberg, New York (1997)
4. Chen, C.X., Zaniolo, C.: SQL^{ST}: A Spatio-temporal Data Model and Query Language. In: Proceedings of the 19th International Conference on Conceptual Modeling, pp. 96–111. Salt Lake City, Utah, USA (2000)
5. Chen, C.X., Wang, H., Zaniolo, C.: Toward Extensible Spatio-temporal Databases: an approach based on User-Defined Aggregates. In: de Caluwe, R., et al.(eds.) Spatio-temporal Databases: Flexible Querying And Reasoning, pp. 55–74. Springer, Berlin, Heidelberg, New York (2004)
6. Erwig, M., Schneider, M.: Spatio-temporal Predicates. In: Transactions on Knowledge and Data Engineering, vol.14, No.4, pp.

881–901. IEEE Educational Activities Department, Piscataway, NY, USA (2002)
7. Forlizzi, L., Guting, R.H., Nardelli, E., Schneider, M.: A Data Model and Data Structures for Moving Objects Databases. In: Proceedings of ACM-SIGMOD International Conference on Management of Data, pp. 319–330. Dallas, Texas, USA (2000)
8. Güting, R.H., Böhlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N.A., Schneider, M., Vazirgiannis, M.: A Foundation for Representing and Querying Moving Object. In: ACM Transactions on Database Systems, Vol.25, No.1, pp. 1–42. ACM Press, New York, NY, USA (2000)
9. Güting, R.H., Böhlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N.A., Nardelli, E., Schneider, M., Viqueira, J.R.R.: Spatio-temporal Models and Languages: An Approach Based on Data Types. In: Sellis, T., et al.(eds.) Spatio-temporal Databases: The CHOROCHRONOS Approach, LNCS 2520, pp. 117–176. Springer, Berlin, Heidelberg, New York (2003)
10. Herring, J.R.(eds.): OpenGIS Implementation Specification for Geographic Information – Simple Feature Access. http://www.opengeospatial.org/standards/sfs (2006). Accessed 30 Aug 2007
11. Khatri, V., Ram, S., Snodgrass, R.T.: On Augmenting Database Design-Support Environments to Capture the Geo-Spatio-temporal Data Semantics. In: Information Systems, Vol.31, No.2, pp. 98–133. Elsevier Science Ltd., Oxford, UK (2006)
12. Laurini, R., Thompson, D.: Fundamentals of Spatial Information Systems. Academic Press, London, UK (1992)
13. Stolze, K.: SQL/MM Spatial – The Standard to Manage Spatial Data in a Relational Database System. In: Datenbanksysteme für Business, Technologie und Web, Tagungsband der 10, pp. 247–264. Leipzig, Germany (2003)
14. Worboys, M.F.: A Unified Model for Spatial and Temporal Information. In: Computer Journal, Vol.37, No.1, pp. 26–34. Oxford University Press, Oxford, UK (1994)

Spatio-temporal Similarity Search

► Top-k Retrieval Techniques in Distributed Sensor Systems

Spatio-temporal Uncertainty

► Moving Object Uncertainty

Spatio-temporal-Authorization-Driven R-Tree

► Geospatial Authorizations, Efficient Enforcement

SPIX

► Data Collection, Reliable Real-Time

Splines

► Constraint Databases and Data Interpolation

Split Designs

► Contraflow for Evacuation Traffic Management

Split, Overlap-Free

► Indexing, X-Tree

SQL

► Oracle Spatial, Raster Data

SQL, Spatial

► PostGIS

SQL/MM

► Oracle Spatial, Geometries
► Spatio-temporal Query Languages

SQL/Temporal

► Spatio-temporal Query Languages

Standards

► Cyberinfrastructure for Spatial Data Integration
► Metadata and Interoperability, Geospatial
► OGC's Open Standards for Geospatial Interoperability

Standards, Consentual

► Spatial Data Transfer Standard (SDTS)

Standards, Critical Evaluation of Remote Sensing

LIPING DI
Center for Spatial Information Science and Systems (CSISS), George Mason University, Fairfax, VA, USA

Synonyms

Earth observation standards; Remote sensing specifications; Remote Sensing Standards

Definition

ISO/IEC Guide 2:1996 defines a standard as a document established by consensus and approved by a recognized body that provides for common and repeated use, rules, guidelines or characteristics for activities or their results aimed at the achievement of the optimum degree of order in a given context. Remote sensing is a method for measuring the properties of an object without contacting the object. Remote sensing standards are those dealing with any aspect of remote sensing, including the data content, metadata, quality, the description of sensors and measuring conditions, etc.

Historical Background

Remote sensing is one of the major methods for collecting geospatial data. The data acquired by remote sensors are called remote-sensing data. Both public and private sectors have collected huge amount of remote sensing data [1]. The data have been widely used in applications ranging from scientific research and military applications to daily socio-economic activities. These applications call for integrated and quantitative analyses of remote sensing data from multiple sources [2]. This requires the interoperability of data from multiple sensors and data producers. In order to achieve the interoperability and integrated analysis, remote sensing standards have to be developed. For several years, since recognizing the importance of standards in the interoperability and sharing of remote sensing data, individual data producers and many national and international organizations have worked on setting remote sensing standards. The most notable organizations setting remote sensing standards include the U.S. Federal Geographic Data Committee (FGDC), which sets the U.S. federal standards on geographic information [3], the Open Geospatial Consortium, Inc. (OGC), which sets the industry specifications for interoperability of geo-processing software and services [4], and the International Organization for Standardization (ISO) Technical Committee 211 (ISO TC 211), which sets the international standards for geographic information [5]. All of these organizations have developed or are developing remote sensing-related standards.

Scientific Fundamentals

Geospatial data, information, knowledge, and services are important resources in an information society. Making such resources easily accessible and useful with various applications is one of the major functions of the geospatial data and information systems. In order to share geospatial resources within the society, those systems have to be interoperable. The key for the success of distributed, interoperable geospatial systems is the standards. Remote sensing is one of the most important sources for geospatial data and information. Therefore, remote-sensing standards are an important set of standards in the family of geographic/geospatial information standards.

According to the principle of information engineering, the remote sensing standards can be classified into four general categories based on the subject a standard tries to address.

- Data
- Processes
- Organizations
- Technology

A remote sensing standard may only address standardization in one standard category or may cover multiple standard categories. Since the discipline of geographic information science mostly concerns the data acquired by remote sensors, the majority of remote sensing standards in the discipline deals with some aspects of remote sensing data to facilitate the sharing and further processing of the data. The major types of remote-sensing standards include the data content, metadata, quality, and encoding.

Remote sensing data content standards provide semantic definitions of a set of objects and their relationships within a remote sensing dataset. The metadata standards provide standardized terminology and definitions for describing the properties of remote sensing datasets. The remote sensing quality standards define the quality requirement, evaluation, and reporting methods. The remote sensing encoding standards define the format in which the remote sensing data are stored.

Normally, a standard approval process is a consensus-building process through which the stake holders of a standard work together to build the consensus. The consensus is documented as the standard. The process of setting a standard is called the standard approval process. Each standard organization has its own documented rules on setting standards. The standard approval process follows those rules. The standard approval processes used by standard-setting organizations are very similar. The process normally involves the development, public comments, and approval stages.

Key Applications

Remote Sensing Standards Developed by U.S. FGDC

The Federal Geographic Data Committee (FGDC) of the United States is an interagency committee, organized in 1990 under the Office of Management and Budget (OMB) Circular A-16 that promotes the coordinated use, sharing, and dissemination of geospatial data on a national basis [6]. The U.S. President Executive Order 12906, dated April 13, 1994, directed FGDC to coordinate the development of the National Spatial Data Infrastructure (NSDI) [7]. The NSDI encompasses policies, standards, and procedures for organizations to cooperatively produce and share geographic data. The 19 federal agencies that make up the FGDC are developing the NSDI in cooperation with organizations from state, local, and tribal governments, the academic community, and the private sector [3]. The FGDC consists of a steering committee, a coordination group, thirteen subcommittees, and fourteen working groups. The subcommittees are organized along the disciplinary boundaries while the working groups deal with issues across the boundaries.

The FGDC develops geospatial data standards for implementing the NSDI, in consultation and cooperation with state, local, and tribal governments, the private sector and academic community, and, to a certain extent, the international community [8]. While the working groups and subcommittees draft the FGDC standards, the FGDC Standard Working Group (SWG) oversees the standard approval processes. Thus far, the FGDC has developed and endorsed over 20 standards since 1995 when the first FGDC standards were endorsed. FGDC standards are mandatory-if-applicable to all federal agencies.

Remote sensing data is one of the major types of geospatial data that the federal agencies collect, archive, distribute, and use. U.S. federal agencies such as NASA, NGA, USGS, and NOAA possess huge amount of remote sensing data. In order to facilitate the interoperability and sharing of remote sensing data, the FGDC SWG set up a NASA-led Imagery subgroup in 1997 to deal with remote-sensing related standards. Before retiring in 2002, the subgroup had developed two FGDC endorsed standards for remote sensing: The FGDC Content Standard for Remote Sensing Swath Data (hereafter referred to as the Swath Standard) [9], and the FGDC Content Standard for Digital Geospatial Metadata-Extensions for Remote Sensing Metadata (hereafter referred to as the Remote Sensing Metadata Standard) [10]. In addition to those two FGDC standards, the FGDC Subcommittee on Base Cartographic Data also developed a remote sensing related standard: The FGDC Content Standard for Digital Orthoimagery [11].

The Swath Standard Swath data is one of many common types of remote sensing data which are produced by scanning and profiling types of remote sensors and provided in the sensor coordinate system without georectification. Most low-level remote sensing data (e. g., NASA EOS data level 2 and lower) are swath data. The major goals of the standard are to 1) provide a common conceptual framework for encoding swath and swath-like data, 2) encourage inter-use of swath and swath-like data through implementation of transfer standards within the conceptual framework, and 3) use participatory involvement in the standard development to reach out to non-federal organizations which will encourage broadly based application of the standard. The Swath Standard defines the swath concept, the minimal components of a swath, and the relationships among the components [9]. The swath standard borrows heavily from the NASA EOSDIS swath concept [12,13,14,15].

The swath standard, classified as a FGDC content standard, provides the standard data model for the swath data. However, it does not standardize the encoding of swath data. In order for a data product to be complied with the standard, the product has to have all mandatory components available in the product with the content defined in the standard, regardless of how the product is encoded and distributed. Thus, this standard provides the foundation for the interoperability and sharing of swath data since it provides a common data model for all federal agencies to follow.

The Remote Sensing Metadata Standard The FGDC Content Standard for Digital Geospatial Metadata (hereafter referred to as the FGDC metadata standard) was the first standard developed and endorsed by FGDC in 1995. The second version of the standard was endorsed by the FGDC in 1998 [16]. The FGDC metadata standard is the most influential standard developed by the FGDC thus far. The standard not only is used by the FGDC metadata clearinghouse for geospatial data discovery, but is also the foundation for the ISO 19115:2003-Geographic Information-Metadata, an official international standard [17]. The metadata standard defines common geospatial metadata that provide information for prospective users to determine the following information about a geospatial data set: its availability, its fitness for an intended use, and the means of accessing and successfully transferring it. Because of its generality, the standard may not meet the metadata needs for specific geospatial domains, including remote sensing. In 1999, the FGDC Imagery subgroup started to develop the remote sensing metadata extensions to the FGDC metadata standard [18]. The objective of the Remote Sensing Metadata Standard is to provide additional information particularly relevant to remote sensing within the framework of the FGDC metadata standard: the geometry of the measurement process, the properties of the measuring instrument, the processing of raw readings into geospatial information, and the distinction between metadata applicable to an entire collection of data and those applicable only to component parts. The Remote Sensing Metadata Standard establishes the names, definitions, and permissible values for new data elements and the compound elements of which they are the components. These new elements are placed within the structure of the FGDC Metadata Standard, allowing the combination of the original standard and the new extensions to be treated as a single entity. The Extensions do not specify either the means by which this information is organized in a computer system for data transfer or the means by which this information is transmitted, communicated, or presented to the user. Endorsed by the FGDC in 2002, the remote sensing metadata standard borrows heavily from the NASA EOSDIS Core System (ECS) metadata model [19] with additions from the photogrammetry community.

The Orthoimagery Standard The FGDC Orthoimagery Standard defines the orthoimagery theme of the digital geospatial data framework as envisioned by the FGDC. The framework will provide a base on which to accurately collect, register, and integrate digital geospatial information. Digital orthoimagery is a part of this basic set of data described as framework data. The standard sets a common baseline that will ensure the widest utility of digital orthoimagery for the user and producer communities through enhanced data sharing and the reduction of redundant data production. The standard stresses complete and accurate reporting of information relating to quality control and standards employed in testing orthoimagery data. This standard describes processing, accuracy, reporting, and applications considerations for NSDI Framework digital orthoimagery, and may be applicable to other data sets which employ the FGDC Framework concepts. In order to support the U.S. geospatial one-stop initiative, the standard is currently being modified to become a U.S. national standard [20].

OGC Specifications

The Open Geospatial Consortium, Inc. (OGC) is a nonprofit international membership-based organization founded in 1994 to address the lack of interoperability in geospatial data, information, and systems. The main work of the OGC is to develop the interoperability specifications through the testbed and consensus-building processes. There are two types of specifications: abstract and implementation ones.

OGC Abstract Specifications The abstract specifications provides the framework or the reference model for geographic information. They contain conceptual models sufficient enough to allow for the creation of implementation specifications. An abstract specification contains at least an essential model of the topic, which in real world terms explains the objects, interfaces, behaviors, and parameters that are the subject of the topic, and an abstract model of the topic, which identifies the classes and subclasses of interest, identifies and describes their relationships, and abstractly specifies the interfaces that are to be implemented in software. Currently, the OGC has completed sixteen abstract specifications. Although all of those specifications are relevant to remote sensing data at various degrees, the most relevant ones to remote sensing are Topic 7: Earth Imagery [21], Topic 15: Imagery Exploitation Services [22], and Topic 16: Image Coordinate Transformation Services [23]. These documents provide the theoretical base for Earth observation and the processing of remote sensing imagery. The OGC modifies the abstract specifications as frequently as needed. Based on the agreement between the OGC and ISO TC 211, the OGC will use relevant ISO standards to replace the existing OGC abstract specifications.

OGC Implementation Specifications The OGC implementation specifications provide unambiguous technology platform specifications for the implementation of standard geo-processing software interfaces. Because major GIS software vendors as well as government geospatial agencies are members of the OGC, those implementation specifications are very influential and are recognized as the international industry standards for geographic information systems. Currently, there are a large number of OGC implementation specifications either endorsed by the OGC or in some stages of the development processes. Among those specifications, the ones most relevant to remote sensing include the Web Coverage Service (WCS) specification [24], the Web Map Service (WMS) specification [25], and the Sensor Markup Language (SensorML) specification [26].

The OGC WCS specification defines the interfaces between web-based clients and servers for accessing online multi-dimensional, multi-temporal geospatial data in an interoperable way. Based on the OGC's definition, all remote sensing images as well as the gridded data, such as DEM, Landuse classification, etc., are part of coverage data. The WCS technology allows data users to specify their data requirements. An OGC compliant WCS server has to preprocess the data on-demand based on users' requirements and returns the data back to users in the form specified by users. In the end, the user will get the data that exactly match their requirements in both the contents and the structure (e. g., format, projection, spatial and temporal coverage, etc.). The WCS version 1.0 was approved by the OGC as an OpenGIS specification in 2003. A new version of the WCS specification, version 1.1, is currently in the OGC specification approval process and expected to become an official specification in 2007.

The OGC WMS specification defines web interfaces for dynamically assembling maps from multiple sources within a heterogeneous distributed computing environment over the Internet. Maps are the visualization of data. A WMS server normally converts the data in its archive to a visualized form (map) based on the requirements from the client. The WMS specification provides a standard way of implementing a data visualization service for remote sensing data. The WMS specification is the first and the widest implemented OGC web service specification. The WMS 1.0 was the first OGC Web-based implementation specification approved in 2000. This version of WMS specification is also approved by ISO as an ISO standard in 2005. The latest version of OGC WMS specification is version 1.3.0, approved in 2006.

SensorML is an OGC best practice document issued by the OGC in 2005 [26]. It provides a model and an XML schema for defining the geometric, dynamic, and observational characteristics of sensors and sensor systems. An OGC best practice document contains a discussion of best practices related to the use and/or implementation of an adopted OGC document and is for release to the public. Best practice documents are an official position of the OGC and thus represent an endorsement of the content of the paper. SensorML was initiated under the auspices of the Committee for Earth Observation Satellites (CEOS) and funded by NASA, EPA, and NGA. It continuously developed under the Sensor Web Enablement (SWE) thread of OGC OWS Initiatives. SensorML provides a functional description of virtually any sensor system including an in-situ or remote sensor on either a stationary or dynamic platform. The SensorML description provides information needed for discovery, processing, and geolocation of sensor observations, [26].

ISO TC 211 Remote Sensing Standards

The ISO TC 211 is a technical committee under the International Organization for Standardization (ISO). It is responsible for setting international standards for geographic information [5]. Since 1997, the ISO TC 211 has been developing international standards in the area of the imagery and gridded data, which includes remote sensing data. From 1997 to 2000, the ISO TC 211 worked on preparation for developing ISO standards through two ISO stage-0 projects: ISO 19121 and ISO 19124. Finished in 1999, ISO 19121-Imagery and Gridded Data project addressed the manner by which TC 211 should handle imagery and gridded data in the context of the field of geographic information [27]. As a result of the ISO 19121 study, the ISO 19124-Imagery and Gridded Data Components was started in 1999 to identify areas of new standards needed for imagery and gridded data. The work also identified aspects of existing parts of the ISO 19100 family of standards that needed to be expanded to address imagery and gridded data. The project was finished in 2000 and produced a review summary that summarized the areas for standardization in imagery and gridded data [28]. As a result of those preparatory efforts, several remote sensing-related standards are being developed in ISO TC 211.

ISO/TS 19101-2 The ISO 19101-2 project, Geographic information – Reference model – Part 2: Imagery, intends to develop an ISO technical specification (ISO/TS) that defines a reference model for standardization in the field of geographic imagery processing [29]. This reference model identifies the scope of the standardization activity being undertaken and the context in which it takes place. The scope includes gridded data with an emphasis on imagery. The ISO/TS 19101-2 defines the imagery reference model using the four viewpoints of Open Distributed Processing (ODP) systems defined in ISO/IEC 10746-2, including enterprise, computational, information, and engineering viewpoints. The reference model will guide the development of imagery-related standards in ISO TC 211 and provide the framework for fitting different imagery-related standards together. Although structured in the context of information technology and information technology standards, ISO/TS 19101-2 is independent of any application development method or technology implementation approach. It is expected that this TS will be published by ISO in 2007.

ISO 19115-2 The ISO 19115-2 project, Geographic information – Metadata – Part 2: Extensions for imagery and gridded data, is developing a metadata standard for imagery (including remote sensing imagery) and gridded data [30]. This International Standard will extend the existing geographic Metadata standard, ISO 19115:2003, by defining the schema required for describing imagery and gridded data. It provides information about the properties of the measuring equipment used to acquire the data, the geometry of the measuring process employed by the equipment, and the production process used to digitize the raw data. This extension deals with metadata needed to describe the derivation of geographic information from raw measurements, including the properties of the measuring system and the numerical methods and computational procedures used in the derivation. The standard is based on the FGDC remote sensing metadata standard with numerous additions and modifications. It is expected that the standard will be available in 2008.

ISO 19128:2005 The ISO 19128:2005, Geographic information – Web map server interface, is an ISO standard that was published in 2005 [31]. It specifies the behavior of a service that produces spatially referenced maps dynamically from geographic information. It specifies operations that retrieve a description of the maps offered by a server, to retrieve a map, and to query a server about features displayed on a map. The standard is based on and almost identical to the OGC WMS specification 1.0.

ISO/TS 19129 The ISO 19129 project, Geographic information – Imagery, gridded and coverage data framework, was started in 2001. The project will develop an international technical specification to standardize concepts for the description and representation of imagery, gridded and coverage data in the context of the ISO 19100 family of standards. The technical specification addresses five areas identified in the ISO 19124 review summary, including data model (or schema), metadata, encoding, services, and spatial registration. The project also serves as a spawning ground for new ISO projects in the related areas [32]. The project was removed by ISO TC 211 in 2004 due to lack of progress and was reintroduced in 2005. It is expected that the technical specification will be available in 2008.

ISO 19130 The ISO 19130 project, Geographic information – Sensor data models for imagery and gridded data, is to develop an ISO standard on sensor and data models for imagery and gridded data. The standard will specify a sensor model describing the physical and geometrical properties of each kind of photogrammetric, remote sensing and other sensors that produces imagery data. It will also define a conceptual data model that specifies, for each kind of sensor, the minimum content requirement and the relationship among the components of the content for the raw data that was measured by the sensor and provided in an instrument-based coordinate system, making it possible to geolocate and analyze the data [33,34,35]. The project was removed by ISO in 2006 due to lack of progress. It is expected that the project will be reinstalled in 2007.

Standards Harmonization

As discussed in the previous sections, remote sensing standards are being developed by different standard-setting organizations. Based on its influence or applicability, a standard could be considered as an agency, federal, national, or international standard. Both OGC and ISO standards are international standards, while FGDC standards are U.S. federal standards. Compared to an ISO standard which is developed by experts from and approved by the member nations, an OGC standard is an international industry standard developed by concerned parties interested in the standard. In order to ensure consistency among the standards developed by different organizations, related standards have to be harmonized. This section will discuss the harmonization efforts among the FGDC, OGC, and ISO TC 211 on remote sensing standards.

The OGC is an official external liaison organization of the ISO TC 211. A cooperative agreement between ISO TC 211 and OGC has been signed to promote the cooperation between the two organizations for developing geographic information standards. The OGC will adopt the relevant ISO 19100 series of standards as its abstract specifications while submitting the implementation specifications to the ISO TC 211 for approval as official international standards. Several OGC implementation specifications have been approved or are being approved as ISO standards, including the ISO 19125-Simple Feature Access [36], ISO 19128-Web Map Server Interface [37], and ISO 19136-Geographic Markup Language [38]. In the remote sensing area, OGC experts have been heavily involved in the development of ISO/TS 19101-2, ISO 19115-2, ISO/TS 19129 and ISO 19130 standards.

The FGDC is a principal member of the OGC and is a sponsor of several OGC interoperability initiatives. The FGDC participates in ISO activities as a member of The InterNational Committee for Information Technology Standards (INCITS) Technical Committee L1. The FGDC remote sensing standards have been tested in those initiatives and served as one of the bases for the development of OGC implementation specifications. Both the FGDC swath and remote sensing metadata standards have been used in OGC Web Service Initiatives for developing the OGC WCS specification and imagery metadata interoperability program report. Because the FGDC is neither an international organization nor an authorized national body representing the U.S. in ISO TC 211, the FGDC cannot directly be involved in ISO TC 211 activities. However, FGDC standards play significant roles in developing ISO standards. Currently, the FGDC swath standard is the basis for the data model part of ISO 19130, and the FGDC remote sensing metadata standard is the basis for ISO 19115-2. In the future, the FGDC will adopt ISO standards as FGDC standards once those standards are adopted as the U.S. national standards. Because of these harmonization efforts, it is likely that the consistency on standards across federal, national, and international levels will be achieved for remote sensing in the near future.

Applications of Remote Sensing Standards

Since the 1990s, many national and international organizations have participated in the development of spatial data and information infrastructures for facilitating the sharing of spatial data and information among broad geospatial data producers and consumers and for supporting geospatial applications in multiple disciplines. Since remote sensing is one of the major methods for acquiring geospatial data, remote sensing standards are always one of the core standards for construction of any spatial data infrastructure. For example, the National Spatial Data Infrastructure (NSDI) initiative of the United States is using the remote sensing standards discussed in this encyclopedia entry for the construction of NSDI [39]. Internationally, the intergovernmental *Group on Earth Observations* (GEO) is leading a worldwide effort to build a Global Earth Observation System of Systems (GEOSS) over the next 10 years [40]. GEOSS will work with and build upon existing national, regional, and international systems to provide comprehensive, coordinated Earth observations from thousands of instruments worldwide, transforming the data they collect into vital information for society [41]. GEOSS is using ISO and OGC standards, including these remote sensing standards discussed in this encyclopedia entry for facilitating the interoperability and sharing [42].

Future Directions

Remote sensing standards are an important part of geographic information standards. Currently, the progress on the development of remote sensing standards is far behind the development of other parts of geographic information standards. It is expected that the OGC and ISO TC 211 will continue to develop remote sensing standards. With the development of ISO/TS 19101-2 and ISO/TS 19129, how the ISO 19100 family of standards will address the standardization requirements on remote sensing will be clearly defined. A series of new ISO standards in remote sensing will be developed with the framework set by ISO/TS 19101-2. Meanwhile, the OGC will concentrate its developmental efforts on the specifications for the sensor web and geospatial imagery processing services. With recent policy change on standard development, the FGDC will concentrate on adopting the remote sensing standards developed by the ISO and OGC rather than developing its own standards. Only on the areas that will not be covered by ISO and OGC, FGDC will develop its standards. Therefore, it is unlikely in the near future that the FGDC will independently develop new remote sensing standards.

Cross References

► Geographic Coverage Standards and Services
► Metadata and Interoperability, Geospatial
► OGC's Open Standards for Geospatial Interoperability

Recommended Reading

1. Asrar, G., Greenstone, R.: MTPE EOS Reference Handbook, NASA Goddard Space Flight Center, Greenbelt, MD (1995)
2. Asrar, G., Dozier, J.: EOS: Science Strategy for the Earth Observing System.American Institute of Physics Press, Woodbury, New York (1994)
3. FGDC, The Federal Geographic Data Committee Homepage. http://www.fgdc.gov. Accessed 1 May 2007
4. OGC, The Open Geospatial Consortium Homepage. http://www.opengeospatial.org. Accessed 29 Apr 2007
5. ISO TC 211. ISO TC 211 homepage. http://www.isotc211.org. Accessed 29 Apr 2007
6. OMB Circular A-16 (2002) http://www.whitehouse.gov/omb/circulars/a016/a016_rev.html. Accessed 10 Sept 2007
7. Executive order 12906 (1994) http://www.fgdc.gov/policyandplanning/excecutive_order. Accessed 9 Sept 2007
8. FGDC Standard Working Group homepage. http://www.fgdc.gov/standards/organization/FGDC-SWG/index_html. Accessed 9 Sept 2007
9. FGDC (1999), Content Standard for Remote Sensing Swath Data, FGDC-STD-009-1999. http://www.fgdc.gov/standards/projects/FGDC-standards-projects/swath_data/index_html. Accessed 9 Sept 2007
10. FGDC (2002), Content Standard for Digital Geospatial Metadata: Extensions for Remote Sensing Metadata, FGDC-STD-012-2002. http://www.fgdc.gov/standards/projects/FGDC-standards-projects/csdgm_rs_ex/rermote-sensing-metadata. Accessed 9 Sept 2007
11. FGDC (1999), Content Standard for Digital Orthoimagery, FGDC-STD-008-1999. Reston, VA. http://www.fgdc.gov/standards/projects/FGDC-standards-projects/orthoimagery/index_html. Accessed 9 Sept 2007
12. Ilg, D., Wynne, D., Fortner, B. (1995) The HDF-EOS Swath Concept. Hughes Applied Information Systems. doi:170-WP-003-001
13. Di, L., Suresh, R., Ilg, D., Meyer, T.: EOSDIS Data Models and Example of Implementations. In: Proceedings of The Third International Conference/Workshop on Integrating GIS and Environmental Modeling (CD-ROM and WWW), Santa Fe, New Mexico, 21–25 Jan 1996
14. Demcsak MF (1997) HDF-EOS Library Users Guide for the ECS Project Volume 2: Function Reference Guide. Hughes Applied Information Systems. doi:170-TP-006-003
15. Wynne, D.L., Klein, L., Gejjagaraguppe, R., Ilg, D. (1997) HDF-EOS Library User's Guide for the ECS Project, Volume 1: Overview and Examples. Hughes Applied Information Systems. doi:170-TP-005-004
16. FGDC (1998) Content Standard for Digital Geospatial Metadata (version 2.0), FGDC-STD-001-1998. Reston, VA. http://www.fgdc.gov/standards/projects/FGDC-standards-projects/metadata/base-metadata/index_html. Accessed 9 Sept 2007
17. ISO 19115:2003. Geographic Information-metadata. http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=26020. Accessed 29 Apr 12007
18. NASA (1999) Proposal for a National Spatial Data Infrastructure Standards Project–FGDC Content Standard for Digital Geospatial Metadata: Extensions for Remote Sensing. http://www.fgdc.gov/standards/projects/FGDC-standards-projects/csdgm_rs_ex/standards/projects/csdgm_rs_ex/csdgm_rs_ex_pro.pdf. Accessed 9 Sept 2007

19. Hughes Applied Information Systems (1994) Proposed ECS Core Metadata Standard, Release 2.0, 420-TP-001-005: Landover, Maryland
20. Geospatial One Stop Standards – Digital Ortho Imagery (2002) http://www.geo-one-stop.gov/Standards/DigitalOrthoImagery/index.html. Accessed 11 Oct 2003
21. OGC (2004), Topic 7: Earth Imagery. https://portal.opengeospatial.org/files/?artifact_id=7467. Accessed 29 Apr 2007
22. OGC (2000), Topic 15:Imagery Exploitation Services. https://portal.opengeospatial.org/files/?artifact_id=7199. Accessed 29 Apr 2007
23. OGC (2000), Topic 16: Image Coordinate Transformation Services. http://portal.opengeospatial.org/files/?artifact_id=7200. Accessed 29 Apr 2007
24. Evans, J. (eds.) (2003) OpenGIS Web Coverage Service(WCS) Implementation Specification. OGC doi:03-065r6. https://portal.opengeospatial.org/files/?artifact_id=3837
25. Beaujardiere, J. de la (eds.) (2006) OpenGIS® Web Maper Server Implementation Specification, OGC doi:06-042 http://portal.opengeospatial.org/files/?artifact_id=14416
26. Botts, M. (eds.) (2005) OpenGIS® Sensor Model Language (SensorML)Implementation Specification, OGC doi:05-086 http://portal.opengeospatial.org/files/?artifact_id=12606
27. ISO/TR 19121: 2000. Geographic Information-Imagery and Gridded data. http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=29775
28. ISO 19124 Review Summary 2001-01. Geographic Information-Imagery and Gridded Data Components. http://www.isotc211.org/protdoc/211n1017/
29. ISO/TS 19101-2. Geographic Information-Reference Model – Part 2: Imagery. http://www.isotc211.org/pow.htm
30. ISO 19115-2. Geographic information – Metadata – Part 2: Extensions for imagery and gridded data. http://www.isotc211.org/pow.htm
31. ISO 19128:2005. Geographic information – Web map server interface. http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=32546
32. ISO 19129. Geographic information – Imagery, gridded and coverage data framework. http://www.isotc211.org/pow.htm
33. ISO 19130. Geographic information – Sensor and data models for imagery and gridded data. http://www.isotc211.org/pow.htm
34. Di, L., Kresse, W., Kobler, B.: The Progress and Current Status of ISO Standard Development Project 19130-Sensor and Data Models for Imagery and Gridded Data. In: Chen, J., Jiang, J. (eds.) The International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences, Vol. 34, Part 2, Commission II, pp. 107-110. International Society for Photogrammetry and Remote Sensing (2002)
35. Di, L., Kresse, W., Kobler, B.: The New ISO TC211 Standard Project on Sensor and Data Model for Imagery and Gridded Data. In: Proceedings of International Geoscience and Remote Sensing Symposium (IGARSS 2001). Sdyney, Australia, 2001
36. ISO 19125:2004. Geographic Information-Simple Feature Access. http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=40114
37. ISO 19128 :2005. Geographic Information-Web Map Server Interface. http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=32546
38. ISO 19136. Geographic Information-Geographic Makeup Language. http://www.isotc211.org/pow.htm
39. FGDC (2004). NSDI Futurer Directions Initiative. http://www.fgdc.gov/policyandplanning/future-directions/reports/FD_Final_Report.pdf. Accessed 29 Apr 2007
40. GEO (2007) About GEO. http://www.earthobservations.org/about/about_GEO.html. Accessed 25 May 2007
41. GEO (2005) The Global Earth Observation System of Systems (GEOSS) 10-Year Implementation Plan. http://www.earthobservations.org/docs/10-Year%20Implementation%20Plan.pdf. Accessed 25 May 2007
42. Christian, E. (2004) The architecture of GEOSS. http://usgeo.gov/docs/GEOSSArchitecture.ppt. Accessed 25 May 2007

Standards, Geographic Data Object (GDO)

► Intergraph: Real Time Operational Geospatial Applications

STAR-Tree

► Indexing Schemes for Multi-dimensional Moving Objects

Static Displays in Constraint Databases

► Visualization of Spatial Constraint Databases

Stationarity

► Hurricane Wind Fields, Multivariate Modeling

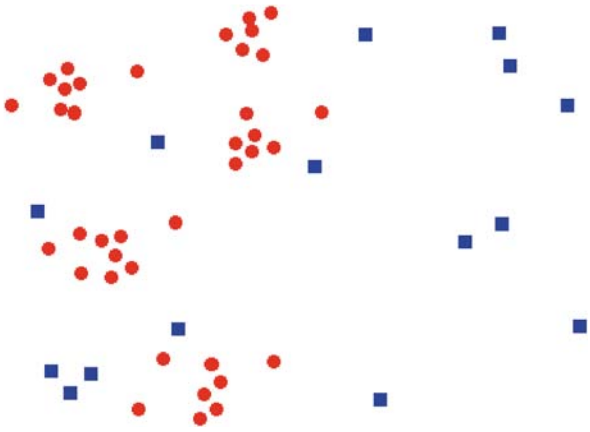
Statistical Descriptions of Spatial Patterns

TEMUULEN TSAGAAN SANKEY
GIS Training and Research Center,
Idaho State University, Pocatello, ID, USA

Definition

Spatial statistics can be defined as a statistical description of spatial data and a spatial pattern or process. Spatial statistics allow a quantitative description along with indications of statistical significance in observational data on a pattern or a process operating in space. This quantitative and statistical description allows the exploration and modeling of spatial patterns and processes and their relationships with other spatial phenomena. Spatial statistics differ from traditional statistics in that spatial statistics integrate space and spatial relationships in the analysis.

Spatial statistics can be used with continuous data as well as three different types of object data: points, lines, and areas [1]. Different spatial statistics can be used depending on the type of data available. Lastly, spatial statistics are closely related to a broader term "spatial analysis", which includes spatial data visualization in a geographic information systems (GIS) environment, spatial data analysis, spatial statistics, and spatial modeling [2]. Spatial data visualization and manipulations help us develop questions and hypotheses about spatial phenomena. Spatial data analysis is used to explore and describe the spatial patterns and processes in question. The spatial patterns and processes can then be formally tested with spatial statistics to determine if the observed patterns and processes are statistically significant. Finally, spatial modeling develops models that predict spatial outcomes. These steps are all closely related and are often difficult to clearly separate.



Statistical Descriptions of Spatial Patterns, Figure 1 A tree seedling distribution map. *Circles* Aspen seedlings, *squares* Douglas-fir seedlings

Historical Background

People have been communicating spatial concepts for millions of years by drawing and reading maps. Spatial statistics, however, are relatively new as they require not only analytical abilities, but also computational skills and computing power. The development of spatial statistical methods has been tightly linked with computer development over the last few decades. When computing power became available along with Fortran codes and punched cards in the late 1960s and early 1970s, spatial statistics and quantitative geography were revolutionized. Many of the spatial statistics techniques including kriging, Ripley's K, and fractal analysis were developed in the 1960s and 1970s [2]. The continuing development of GIS over the last 40 years and the regeneration of other software and extensions have enabled wider uses of spatial statistics. Most GIS, remote sensing, and statistical software programs now have spatial statistical analysis functions and tools that are simple to use. Another related advancement in recent decades is the increasingly abundant availability of spatial data, such as remote sensing data acquired by satellites. This allows the more common use of spatial statistics in various fields of research, where quantitative skills have become more important.

Scientific Fundamentals

Spatial statistics are used to quantify a pattern or process. Many maps and visualization tools might show obvious patterns in the data. For example, a tree seedling distribution map might show a clustered pattern in aspen, but a random or uniform pattern in Douglas fir (Fig. 1). However, sometimes spatial patterns are not obvious or it is unknown if the observed patterns are statistically significant. In these cases, spatial statistics can be used to reveal the less obvious patterns and to determine if the casually observed patterns such as clusters really exist and are statistically significant. Spatial statistics help identify the geographic distribution of a feature, determine whether a particular spatial pattern or a directional trend exists in a feature or a data layer, identify where exactly the patterns and trends exist, and what other features these patterns and trends are related to [3]. For example, a tree seedling distribution might be clustered and the clusters might be found nearby a seed source.

Statistical analysis of spatial data faces several fundamental issues including scale differences, the modifiable areal unit problem, and edge effects. One particularly important issue is that spatial data are usually not samples. Indeed, spatial data often represent the whole population of interest or the entire study region. Spatial data, therefore, violate the basic assumptions of traditional statistics. We deal with this issue by recognizing a spatial pattern as one of many possible realizations that could result from a spatial process. Spatial statistics then formally tests if the realization, or an observed spatial pattern, indeed could be a result of the particular spatial process in question or if the realization is a result of complete spatial randomness (also known as independent random process), which is the null hypothesis [1,2,3].

Spatial processes and patterns can be statistically analyzed using various computer software programs. Currently, some spatial statistical analysis such as spatial autocorrelation tests can be completed in a GIS environment. However, the statistical tests that can be conducted in GIS are often fairly limited. For example, if we want to know if there is a statistically significant association between a set of points that represent tree seedling establishment and a set of polygons that represent patches of burned forest, a statistical test can not be readily performed with the ArcGIS software. The ArcGIS package is not intended for complete spatial statistical analysis and spatial modeling tasks. Therefore, one often needs to incorporate the available ArcGIS tools with other sources of software by exporting the GIS data to other data formats usable with other software. It is also possible to use "macrolanguage" facilities from the GIS environment to access other programs that conduct spatial statistical tests. In this case, one does not need to leave the GIS environment to access spatial statistics tools. Common statistical packages including R, MINITAB, and S-plus have spatial statistical analysis tools and some have a built-in interface with a GIS environment. In addition, some of the digital image-processing software programs such as IDRISI now have built-in spatial statistical tools.

Spatial statistics can be used for two general purposes. Firstly, spatial statistics are concerned with determining and predicting the attribute values in a given variable [1]. In this case, the objective is to describe the observed values at a few locations and predict other values at unmeasured locations. For example, using measured amounts of precipitation at a few locations and their observed variation in space, precipitation at other unmeasured locations can be predicted. Common methods used for this purpose include spatial autocorrelation tests and interpolation approaches such as kriging. They are often used with continuous data. Secondly, spatial statistics are concerned with detecting a spatial pattern such as clusters or a directional trend in a given variable or a feature. In this case, the objective is to reveal a spatial pattern in the data and explain the observed pattern as related to other spatial phenomena. For example, epidemic disease cases might be clustered around a pollution source. The variable of interest here is the spatial location and spatial arrangement of the feature (i. e., the disease cases). Spatial statistics used for this purpose are often descriptive and can be used with three different types of object data: points, lines, and areas.

A spatial process is called stationary or homogenous if the statistical properties of the attribute values are independent of location [1]. In other words, the mean and variance of observed attribute values at different locations across the study region are constant. For example, precipitation might not vary across a homogenous terrain or a small study area. In contrast, a spatial process is nonstationary or heterogeneous if the mean, variance, and covariance of observed attribute values change across a study region [1]. In this case, precipitation amount, for example, changes from one location to another. Spatial patterns often express spatial dependence in two different ways. First, the probability of a spatial phenomenon occurring at each location varies throughout a study region. This is known as a first-order effect of a spatial process. Second, the occurrence of an event at one location increases the probability of the same event occurrence at the neighboring location, sometimes resulting in clusters across a study region. This is known as a second-order effect. Real world spatial patterns are often a result of a combination of first- and second-order effects. Some spatial statistical methods are concerned with first-order effects, while others describe second-order effects.

Continuous Data Analysis

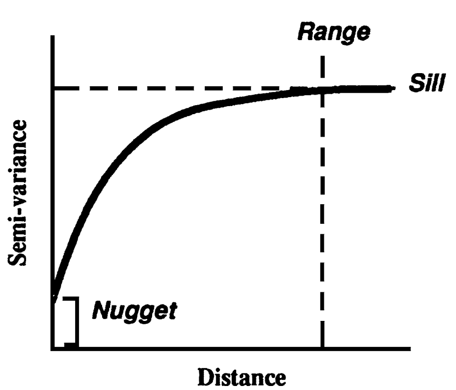
Some spatial statistics deal with data on spatially continuous variables observed at specific locations across a region [1]. Examples of such data include air temperature, precipitation, and soil moisture. The main objective of such statistics is often to describe the spatial variation in the observed values and then to predict the values at unsampled locations. The result of such statistics can be a smooth surface that describes global trends or first-order effects. Another result can be an interpolation of values at unsampled locations close to sampled locations or a description of second-order effects. Commonly used methods include spatial moving averages, spline smoothing, triangulated irregular network (TIN), and kriging [1,4]. Spatial moving averages take n number of nearest sampled points and average the sample values at those points to estimate the value at the location with no recorded value. The estimated average is not weighted, in that it does not take into consideration the relative distances of the nearest sampled points. An extension of this method is an application of a fixed radius, where a specified distance from the unsampled point is used to decide which points contribute and which do not contribute to the estimate of the average. A further extension of this method is inverse distance weighting, where weights (ω) are applied to the distance (d) of each point that contributes to the estimated average [2]. The applied weights are proportional to the inverse of the distance of each point to the point to be interpolated. This allows the points closer to the interpolated point to get more weight than points farther away:

$$\omega = \frac{1/d_{ji}}{\sum_{i=1}^{m} 1/d_{ji}} .$$

TIN methods use n number of locations, each of which gets an associated "territory" or polygon assigned, a process called Dirichlet tessellation. The resulting territories are known as Voronoi or Thiessen polygons. In a GIS environment, Thiessen polygons are created such that each polygon is assumed to have a different value associated with it. The joins of neighboring polygons are not smoothed and can exhibit abrupt differences. This method, therefore, does not provide a smooth surface across the study region. These methods are based on the fundamental geographic idea that the values of a given spatial variable are more likely to be similar at closer locations than at distant locations. In other words, spatial data can often be correlated with itself at short distances. This phenomenon is known as spatial autocorrelation [1,2,3,5,6]. The degree to which the data is spatially dependent can be described in a semivariogram. A semivariogram is based on the idea of a variogram, which is a plot that shows differences in observed attribute values against distances (d) between all pairs or a specified number of pairs of locations. The cloud of points in this plot can be organized into bins of distance groups, which are known as lags. Each lag has an associated mean value. The lags plotted against their mean values indicate whether closer locations indeed have similar attribute values and whether differences in mean attribute values change as distance increases. When the squared differences in observed attribute values are plotted in a similar manner against the distances between all possible pairs of locations, the plot is called a semivariogram. Thus a semivariogram can be expressed as:

$$2\hat{\gamma}(d) = \frac{1}{n(d)} \sum_{dij=d} (z_i - z_j)^2$$

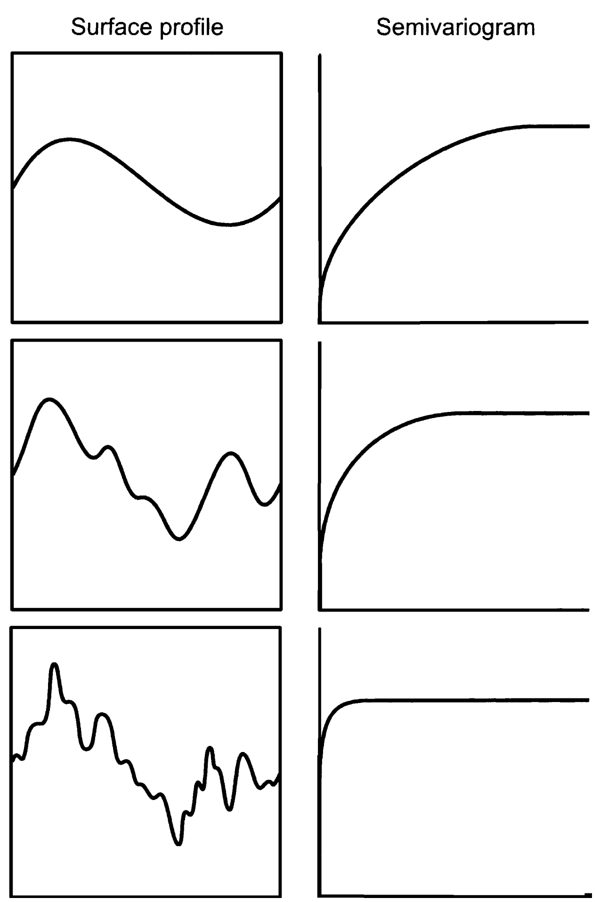
where $\hat{\gamma}$ represents the semivariogram, d is distance between pairs of points, n is the number of points, and z_{ij} are the observed values [2]. In a semivariogram, the variance at zero distance is often greater than zero due to noise in the data and is called a nugget effect. The distance at which the semivariogram plateaus is known as range. There is no spatial dependence beyond the range and the value becomes constant. This constant value is termed the sill [1,2] (Fig. 2).



Statistical Descriptions of Spatial Patterns, Figure 2 A semivariogram

A method that combines inverse distance weighting and the information from a semivariogram is kriging [2]. Kriging is an interpolation method that uses the observed values as sample data and determines the optimum combination of weights to be used in estimating the values at points with no recorded data [1]. Kriging first describes the spatial variation in the observed variable, summarizes the spatial variation in a mathematical function, and then uses this function to determine the weights to be used in the interpolation. The spatial variation of the variable of interest is described by a semivariogram. The semivariogram is then summarized using a mathematical function, which is often fitted by trial and error or by eye [2] (Fig. 3). This should be based on a solid understanding of the spatial variable in question. The default function (e. g., a spherical model) of the GIS software may not be the best choice for many spatial variables.

Other methods used in testing for second-order effects include Moran's I and Geary's C [2,5,7]. These methods can be used with all kinds of spatial data including object



Statistical Descriptions of Spatial Patterns, Figure 3 Common surface profiles and fitted semivariograms [1] data of points, lines, and areas. Moran's I is a measure of correlation in spatial data and can be expressed as:

$$I = \frac{n}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}(y_i - \bar{y})(y_j - \bar{y})}{\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}}$$

where y is the observed value at each location and $\bar{y}$ is the mean value, n is the number of points, w_{ij} is the weight associated with each value to determine which points are adjacent (or close) to be correlated with the point in question. Moran's I can test for spatial autocorrelation in the observed attribute values as well as the spatial locations of the variable in question. Moran's I values range from –1 to +1 and indicate if the observed spatial pattern is clustered, random, or uniform. Moran's I values closer to +1 indicate clustering, while values closer to –1 indicate uniform patterns. Values closer to 0 indicate random patterns.

Geary's C is similar to Moran's I and is expressed as:

$$C = \frac{n-1}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}(y_i - y_j)^2}{2\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}}$$

with the same notations as Moran's I above. The interpretation of the results, however, is different. Geary's C values of 1 indicate no spatial autocorrelation, while values less than 1 indicate positive spatial autocorrelation. Values greater than 1 indicate negative spatial autocorrelation.

Point Data Analysis

Point patterns include a series of point locations, where a spatial phenomenon of interest, or an event, occurred. Events could for instance be tree seedlings of a particular species found throughout a forested region. Descriptive methods of point patterns include density estimates, quadrat counts, kernel window, G, F, and K functions, and nearest-neighbor distance. Point patterns described by these methods then can be formally tested against complete spatial randomness to determine if the observed pattern is unusual and statistically significant. Complete spatial randomness can be computer-generated a number of times (100 times, 1,000 times, etc.) to meet the required n using simulation models [4]. The test indicates if the observed point pattern is significant and if the points follow a Poisson distribution [4]. The test can also indicate if the observed pattern is clustered, random, or uniform. The Poisson distribution assumes no first-order effects and no second-order effects. Other methods such as the heterogeneous Poisson process, the Cox process, and the Poisson cluster process can be used to model clustering [4]. To model regular point patterns, a simple inhibition process or Markov point process can be used.

Quadrat count is an approach where the study region is divided into equal-sized quadrats within which the number of events is counted to characterize the spatial pattern across the entire study region [4]. The number of events counted in each quadrat is divided by the area of each quadrat to estimate abundance or intensity. This estimate is then used to construct a frequency distribution. Quadrat counts can also be used with various indices such as David and Moore's index of clumping, Lloyd's index of patchiness, and Morisita's index of dispersion to determine if a point pattern is clustered, regular or random [4]. A similar concept to quadrat count is Kernel window, which is a moving window of a specified size that estimates probability density and provides a smooth histogram.

While quadrat count and kernel window describe first-order properties of point patterns throughout the entire study region, nearest-neighbor distances describe second-order properties or the relationship between individual events within a local area. There are two different types of nearest-neighbor distances [1,2]. The first is a distance between a randomly chosen event and its nearest neighbor. The second is a distance between a randomly chosen location and the nearest event to this location. Nearest-neighbor distances are used to construct a probability distribution which can be used to examine spatial dependence at a local scale. A probability distribution can be estimated by a G function or F function, which are similar and can be expressed as the following for a random point process [2]:

$$E[G(d)] = 1 - \mathrm{e}^{-\lambda \pi d^2}$$

$$E[F(d)] = 1 - \mathrm{e}^{-\lambda \pi d^2}$$

where E is the expected value for mean number of events within nearest neighbor distance d, λ is intensity, and $e \approx 2.7182818$, the base of the natural logarithm system. The expected G and F functions can then be compared to the observed functions to determine if the observed pattern is unusual and statistically significant. Similarly, the K function also measures second-order effects and estimates the mean number of events within a distance d of an arbitrary event. It can be defined as:

$$E[K(d)] = \pi d^2$$

where E is the expected value for the mean number of events within distance d.

Line Data Analysis

Properties of line data include length (or distance), direction, and connection. Line data are more difficult to store, represent, and analyze compared to point data. In a GIS

S

environment, lines are often stored as a sequence of points that are connected by segments. The length of a line could be very different depending on the scale and resolution at which it is represented. A mathematical approach called fractal dimension can be used to address this issue. Fractal dimension measures the length of a line at different resolutions to determine how it changes from one resolution to another. More complex lines have higher fractal dimensions [2].

Statistical analysis is not commonly applied to line data because a null hypothesis (e. g., complete spatial randomness) is difficult to develop for line data [2]. Simple descriptive statistics of line data include mean length and standard deviation, path density, path frequency, and path maintenance. Formal statistical tests can be applied to directional data to compare the observed direction to random directions. Formal tests are also used with trees and graphs [2]. A tree is a common line pattern, where connected lines generate a branched tree with no closed loops. A stream ordering approach is used with tree networks where an order is assigned to each leaf and is increased by one at each junction. A graph is a network of lines where closed loops can be present. A mathematical approach, called a graph theory, has been developed to describe graphs. Graphs are topological and describe only connectivity and adjacency, which can also be a matrix.

Area Data Analysis

Areas are more complex than the other two types of object data. Their properties include two-dimensional area, skeleton, centroid, shape, spatial pattern, and fragmentation. Spatial statistics of area data are concerned with detecting a spatial pattern or trend in attribute values of a group of areas. Common methods include proximity measures, spatial moving averages, median polish, and kernel estimation [1]. Proximity measures can be a simple distance measure between centroids of areas or a combination of distances between centroids and other attributes such as shared boundary length. Proximity measures can also be used in a spatial proximity matrix form.

Spatial moving averages estimate the mean of the attribute values in neighboring areas. Here the spatial moving average is used with a different objective compared to the spatial moving average used in interpolation. The objective here is to smooth the attribute values in a neighborhood, so that a global trend or first-order effects can be easily described. Spatial proximity matrix can also be used with a spatial moving average to determine which areas should be and which areas should not be included in estimating the average. Median polish is a similar method that can be used with a regular grid data. As implied by the name, this method uses medians rather than the mean to describe a global trend in the attribute values. It is, therefore, less sensitive to outliers and provides a smooth surface.

Kernel estimation can be used with area data to describe first-order properties. This method can be applied as a moving window of a specified size to estimate probability density and to provide a smooth histogram. Here the application of kernel estimation is the same as the approach used with continuous data. The attribute value at the centroid of each area is used instead of the values associated with points as in continuous data. Kernel estimation, unfortunately, is not often used with other geometric properties of area data.

The above described methods are used to describe first-order properties of area data. Second-order properties of area data can be described with spatial autocorrelation tests. Both Moran's I and Geary's C can be used with area data. Here the attribute values of areas and the centroids of areas can be tested for spatial autocorrelation [1]. Another method that can be used is the joins count test for spatial autocorrelation in area data. Joins count is the number of times each possible join is observed in a dataset. An observed joins count can be compared to complete spatial randomness or expected joins count of an independent random process. The observed joins count can then be formally tested for statistical significance.

Key Applications

Environmental Science

Many applications of spatial statistics are used in environmental sciences. Plant ecologists study spatial patterns of individual plants and plant communities as related to ecological processes such as competition, disturbance, and regeneration mechanisms. Spatial patterns of plants can also be associated with water and nutrient availability as impacted by different land management options. At a landscape scale, patterns of land cover types and disturbance regimes can be analyzed using remote sensing data and other GIS data layers.

Earth Science

Geologists predict the spatial distribution of mineral deposits and estimate the amount of mineral resources, water, and fossil fuels using interpolation methods. Seismologists use spatial statistical analysis to determine if earthquakes have a spatial pattern associated with geological features such as faults and to determine their predictability. Soil scientists use spatial statistics to describe the distribution of soil characteristics, including properties that are important for agriculture, engineering, and remediation.

Public Health

Epidemiologists study spatial patterns of diseases to determine if disease distributions are associated with possible sources of pollution such as water pollution sources, air pollution sources, and industrial activities. They also describe the spread of epidemic diseases and predict the rates and distribution of epidemic diseases.

Marketing and Economics

Spatial statistics can be used to determine optimum areas for marketing goods and services. Analyzing current distributions of retail stores, for example, is important in determining locations of new stores and services. Spatial statistics can also be used to identify clusters of people of certain age groups or interests to determine optimum locations for new stores and services.

Urban Planning and Management

Urban planners can use spatial statistics to decide on zoning regulations and determine optimum areas of new development and growth. Analysis of current land use, planning conflict, and development patterns help develop new zoning and identify priority areas for transportation and infrastructure development. Spatial statistics can also help predict future development patterns and scenarios.

Future Directions

Increasing computing power and increasingly abundant digital spatial data in the recent years have changed the role of GIS and spatial statistical analysis. An emerging research field, geocomputation, advocates newly developing, computationally intensive tools including artificial neural networks, agent models, and cellular automata in spatial data analysis. As a modeling and analysis toolbox for interactive and complex systems, geocomputation aims to investigate dynamic nondeterministic systems, human and natural, in their spatial context by bridging the gap between geography and computer science. Geocomputation includes sophisticated spatial analysis and classification techniques that have several advantages over traditional methods. These advantages include the ability to deal with high-dimensional and noisy data and versatile combinations of statistical methods.

Cross References

► Data Analysis, Spatial
► Spatial and Geographically Weighted Regression

Recommended Reading

1. Bailey, T.C., Gatrell, A.C.: Interactive Spatial Data Analysis. Prentice Hall, Harlow, UK (1995)
2. O'Sullivan, D., Unwin, D.J.: Geographic Information Analysis. Wiley, Hoboken, NJ, USA (2003)
3. Mitchell, A.: The ESRI Guide to GIS Analysis. Vol. 2: Spatial Measurements and Statistics. ESRI, Redlands, CA, USA (2005)
4. Ripley, B.D.: Spatial Statistics. Wiley, New York, USA (1981)
5. Longley, P., Batty, M.: Spatial Analysis: Modeling in a GIS Environment. Wiley, Cambridge, UK (1996)
6. Cliff, A.D., Ord, J.K.: Spatial Autocorrelation. Pion, International Co-Operative Publishing House, London, UK (1973)
7. Cormack, R.M., Ord, J.K.: Spatial and Temporal Analysis in Ecology. International Co-Operative Publishing House, Fairfield, USA (1979)

Statistical Space-Time Modeling

► Hurricane Wind Fields, Multivariate Modeling

Statistical Techniques

► Crime Mapping and Analysis

Statistics, Spatial

► Local and Global Spatial Statistics

Stereotypes

► Unified Modeling Language Extension Mechanisms

S

Storage Utilization

► R*-tree

Stream Data Mining

► Stream Mining, Spatial

Stream Mining, Spatial

Arnold P. Boedihardjo
Department of Computer Science, Virginia Polytechnic Institute and State University, Falls Church, VA, USA

Synonyms

Stream data mining; Spatio-temporal data mining

Definition

Spatial stream mining is the process of discovering novel patterns, rules, and trends within a set of spatial streams. A spatial data stream is characterized as a data stream which possesses both spatial and non-spatial attributes. Examples of spatial data streams are the location of a continuously moving vehicle in a time period and the non-spatial measurements of a geospatially aware sensor network. In general, the data stream is "continuous, mutable, ordered, fast, high-dimensional, and unbounded" [2]. Spatial stream mining hence focuses on developing and optimizing mining techniques for spatial data streams.

Historical Background

Active work in stream query processing began in early 2000 with systems such as STREAM [1] and COUGAR [16]. Most of these efforts placed heavy emphasis on query processing and gave little attention to data mining tasks. Then later much interest arose to answer some of the fundamental mining tasks in data streams. Tasks such as classification were addressed by Domingos and Hulten's development of the very-fast-decision-tree (VFDT) [4]. However, the focus was on general multidimensional streams and it would be later that the research community deepened their specialty to include multimedia data streams. Many of the early works in spatial stream mining were derived from the multimedia field. A natural application of spatial mining was to perform image analysis, a subfield of multimedia. Early spatial stream works include efforts such as the classification and detection of outliers in stream satellite images [10,11,15].

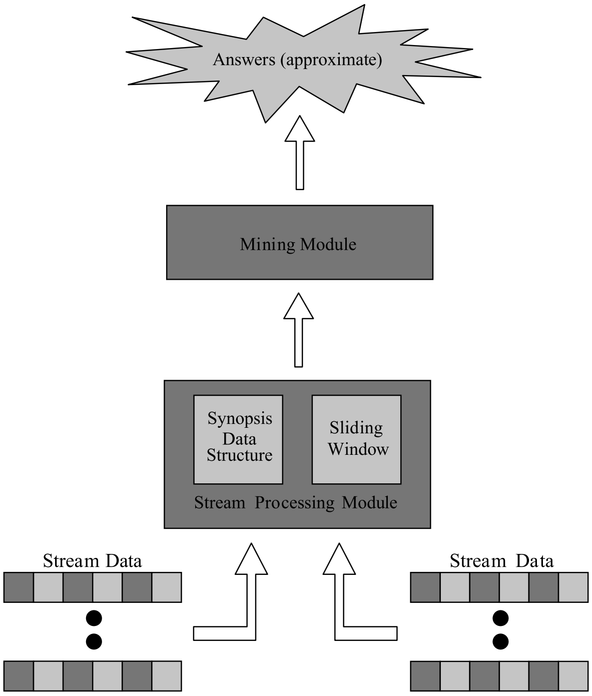
Scientific Fundamentals

Since spatial stream mining share many commonalities with general data stream mining, we begin by stating the key goals of a general data stream mining system (DSMS) as follows [2,9]:

1. *Fast processing of fundamental queries* – enable efficient processing of primitive queries on the data stream necessary for the mining task. An example is the join query, where joined results between multiple data streams are used as precursors to perform a particular mining task.
2. *Efficient mining of patterns, rules, or trends* – provide a timely and compact procedure to compute results of the defined mining task.

The SDMS subject to following constraints:

1. *Small memory space*
2. *Limited computational resources*
3. *Short time window*



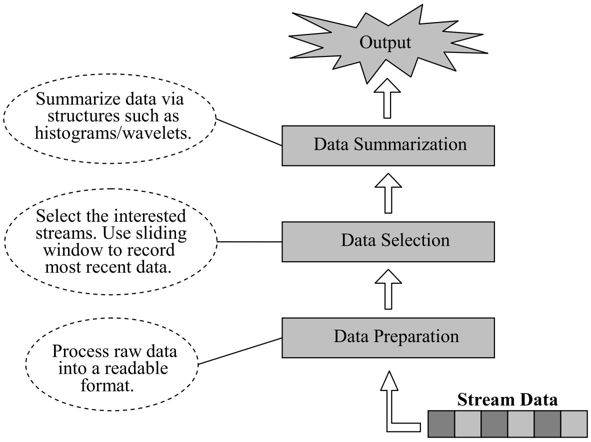
Stream Mining, Spatial, Figure 1 Spatial data stream mining system archetype

A spatial data stream mining system (SDSMS) naturally follows the goals and requirements of DSMS. Because SDSMS must process potentially complex spatial objects, adherence to the above constraints is a key research issue. The properties and requirements of DSMS imply that the stream mining algorithm should possess an asymptotic time-complexity that is linear to the data size. A common mitigating strategy of DSMS is to produce approximate results which in most cases are sufficient to answer the data mining task. Figure 1 illustrates the architecture and components of SDSMS.

In the general SDSMS framework (Fig. 1) incoming data streams arrive at the *stream processing module*. Summarized statistics and direct access to a subset of the stream information (stored in the form of a sliding window) are computed and made available to the mining module. The *mining module* employs the forwarded information to perform specific mining tasks (e. g., classification). The following subsections elucidate the key concepts of the *stream processing* and *mining* modules.

Stream Processing Module

This stream processing module (Fig. 2) is responsible for the *data preparation*, *selection*, and *summarizations*. Below we describe these three components in detail.

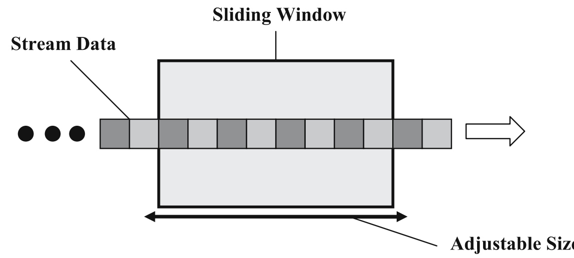


Stream Mining, Spatial, Figure 2 Stream processing module

Data Preparation Data preparation includes data formatting and cleaning. Incoming spatial data (especially images) arrive in raw formats (e. g., bitmaps) which in many cases cannot be efficiently handled by the SDSMS. Hence, efficient methods must be used to transform the data into formats which can be rapidly processed by the data selection and summarization subsystems. Data cleaning is necessary to reduce errors and potential computational overhead. A typical data cleaning task would be to remove invalid data items (e. g., remove negative values when non-negative values are expected) using a filtering-based method.

Data Selection The data selection task decides on how much of the data stream will be stored in memory. A general data structure used for storing the data elements is an adjustable sliding window (Fig. 3). The window will record the most recent data items up to the size of the window. Data replacement policy and window size readjustments will depend on the characteristics of the application. For example, a static size and First-In-Last-Out replacement policy window will be suitable for constant rate data stream, but may not be appropriate for handling bursty data. One important issue for this task is the handling of multiple streams with different and varying incoming rates. A single stream may arrive at a high rate, bombarding system resources that prevent other streams from being processed. Data selection must be able to mitigate a wide spectrum of stream arrival characteristics with the limited space and computational resources.

Data Summarizations The data summarization task provides updated statistical information about the overall data populations through synopsis structures such as his-



Stream Mining, Spatial, Figure 3 Adjustable sliding window

tograms and wavelets coefficient trees. Several other summarization approaches exist, especially ones developed from compression analysis, but these techniques must be adapted to meet the SDSMS constraints. Therefore, it must minimally expend the system's computational and space resources. This can be done by employing iterative update techniques and allowing higher synopsis information error.

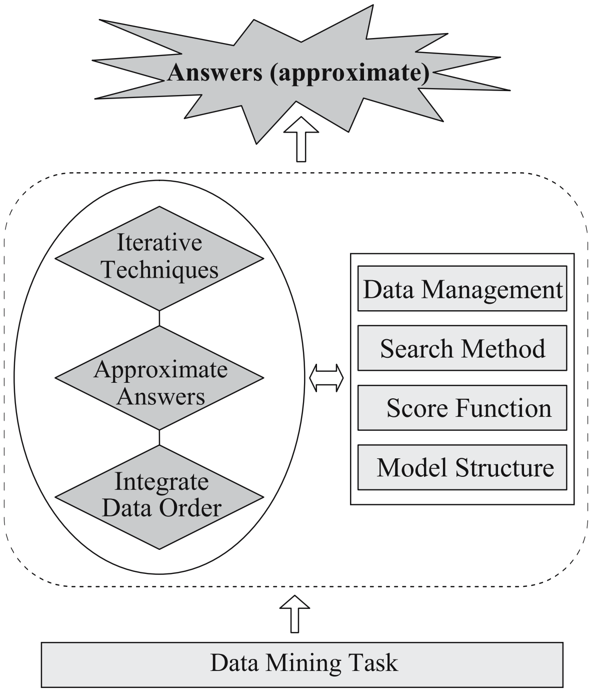
Mining Module

The mining module will process the window records and synopsis information to determine the results of the assigned mining query. Because of the SDSMS's highly restrictive constraints, different mining tasks will often necessitate specific mining approaches. Generalized mining approaches can seldom fulfill the constraints of data stream environments. The mining tasks defined in the SDSMS mining module are similar to the classical spatial mining tasks but with the following key differences:

1. ***Iterative updates*** – *Employ iterative updates to the model*
2. ***Approximate answers*** – *Obtain approximate answers rather than determining exact solutions*
3. ***Data ordering*** – *Integrate data order information*

The first two adaptations are done to achieve single pass data examination, lower computational overhead, and reduce memory usage. The third, however, is an inherent property of data streams. Since data arrive in order, the ordering within the context of the mining task needs to be considered. Data ordering usually manifests in the form of timestamp information.

A data mining process is composed of the following components: data mining task, model structure, score function, search method, and data management technique [5]. For stream data, the model structure, search method, score function, and data management technique may all need to be altered in order to fulfill the SDSMS constraints. However, since an adaptation to utilize efficient iterative update techniques and approximate results may suffice the SDSMS requirements, changes incurred to the data min-



Stream Mining, Spatial, Figure 4 Mining module architecture

ing algorithm can be limited to the search method and data management technique. Excellent examples of these adaptations can be found in [4,7,8,13]. Although dramatic changes to classical data mining algorithms can be performed to meet the stringent SDSMS criteria, they can be unnecessary and do not provide better results than modifications of a few algorithmic components.

Due to the generation of approximate answers, the mining component must address all aspects of information uncertainty.. Uncertainty may arise from the stream processing module or the mining algorithm itself. In either case, the mining module must be able to quantify the uncertainty of results and if possible, parameterize the error bounds. In general, data uncertainty is accumulated from the stream processing module due to the limited window size and the lossy compression nature of the summarization statistics. Errors in raw data (a common occurrence for many applications) are external to the SDSMS; however, the window and summarization statistics can be designed to reduce further errors. For example, increasing the window size can lower data errors; therefore, we can increase the window size by allowing a lossless compression technique to be used over the window records. Additionally, summarization statistics can be tuned to reduce errors by integrating expert knowledge of stream behavior. For example, a synopsis using Wavelet Transform [3] may provide better accuracy than Discrete Cosine Transform [12].

Aside from the technical challenges mentioned above which are primarily derived from the SDSMS constraints, a separate but crucial issue that must be addressed is the management of *concept drifts*. For several SDSMS applications, incoming data streams will exhibit high amounts of changes in the underlying data generator. The mining module must therefore recognize (detect) these changes and make the appropriate alteration to its model structure. For example, classification tasks would be greatly enhanced if it can detect or adapt to concept drifts when they occur. In classification, concept drifts can sometime render the model structure useless. Therefore, if the SDSMS mining module is able to detect these concept changes, then it can be instructed to incrementally update the current model.

Key Applications

The applications of SDSMS can be classified into the following groups: (1) streams of data composed of varying spatial attributes and non-spatial attributes and (2) streams of data comprised of static spatial attributes and varying non-spatial attributes. In the first group, mining is directly performed on the spatial attributes. In contrast, the second group of applications invokes the mining tasks on the varying non-spatial attributes and treat the spatial attributes as secondary constraints. Image analysis and moving objects can be regarded as domains falling under the first group. Geospatially aware sensor networks can be classified into the second group. We give further descriptions and examples below.

Image Analysis

SDSMS techniques are used to discover patterns, rules, and anomalies in a set of image data. The range of data types can vary from batched source of aerial images to streamed content of real-time videos. A large portion of the applications are aimed for surveillance tasks. For example, classification tasks are used on aerial images to label certain geographical features (e. g., mountains, lakes, etc) while anomaly detection techniques are used to track natural events such as hurricanes [17]. The highway transportation domain provides another example of SDSMS techniques used for real-time surveillance. Using the highway video cameras, one can employ both pattern recognition and anomaly detection approaches to automatically recognize emergent events such as vehicular collisions [14].

Continuous Queries of Moving Objects

SDSMS approaches are applied to moving objects for determining transient and reoccurring patterns. An example of this application is the convex hull detection which can maintain and determine convex-shaped clusters [6]. Anomaly detections can also be performed on moving objects by using SDSMS methodologies. An important use of anomaly detection is to find those objects that diverge from expected trajectories. This feature has applicability for specific domains, such as the monitoring of commercial airline flights. Divergent (hence anomalous) trajectories can serve as actionable clues to an emergent event.

Geospatially Aware Sensor Networks

Nodes in a sensor network possess information of their locations (unchanging spatial attributes for the most part) and include updates of their non-spatial measurements (e. g., temperature). The non-spatial attributes are essentially time-series values. Mining tasks such clustering may need to be performed on these time-series data to develop models for the network. As an example in the transportation network, clustering can be applied to monitor congestion spreading. However, since several nodes can exist within the network, clustering *all* of the nodes' time-series information may prove to be too cumbersome for use in a real-time environment. Using the nodes' spatial information and recognizing their spatial dependence, one can drastically prune many of the nodes' time-series information to reduce the data to a size that will satisfy the real-time requirement.

Future Directions

The amount of spatial data is growing rapidly and spatial stream mining provides the framework by which to manage this large and expanding data. The future of SDSMS will further delve into four fronts: (1) research theoretical fundamental issues which are necessary for providing a deeper insight to solving larger classes of spatial stream mining tasks, (2) deepen investigations into classical mining tasks to provide their respective adaptations for data streams, and (3) determine other problem domains for which SDSMS can be applied.

Cross References

► Autocorrelation, Spatial
► Biomedical Data Mining, Spatial
► Data Analysis, Spatial
► Image Mining, Spatial
► Outlier Detection, Spatial

Recommended Reading

1. Arasu, A., Babcock, B., Babu, S., Datar, M., Ito, K., Nishizawa, I., Rosenstein, J., Widom, J.: STREAM: The Stanford stream data manager. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, California, 2003, p. 665
2. Babcock, G., Babu, S., Datar, M., Motwani, R., Widom, J.: Models and issues in data stream systems. In: ACM Symposium on Principles of Database Systems (PODS), Madison, Wisconsin, 2002, pp. 1–16
3. Chakrabarti, C., Vishwanath, M., Owens, R.: Architectures for wavelet transforms: a Survey. J. VLSI Signal Process, pp. 171–192 (2005)
4. Domingos, P., Hulten, G.: Mining high-speed data streams. In: Proceedings of ACM Special Interest Group on Knowledge Discovery and Data Mining, Boston, Massachusetts, 2000, pp. 71–80
5. Hand, D. J., Mannila, H., Smyth, P.: Principles of Data Mining. MIT Press, Cambridge, MA (2001)
6. Hershberger, J., Shrivastava, N., Suri, S.: Cluster hulls: a technique for summarizing spatial data streams. In: Proceedings of IEEE International Conference on Data Engineering, Atlanta, Georgia, 2006, p. 138
7. Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: Proceedings of ACM Special Interest Group on Knowledge Discovery and Data Mining, San Francisco, 2001, pp. 97–106
8. Natwichai J., Li, X.: Knowledge maintenance on data streams with concept drifting: International Symposium on Computation and Information Sciences (CIS), Shanghai, China, 2004, pp. 705–710
9. O'Callaghan, L., Mishra, N., Meyerson, A., Guha, S., Motwani, R.: Streaming-data algorithms for high-quality clustering. In: Proceedings of IEEE International Conference on Data Engineering, San Jose, California, 2002, pp. 685–694
10. Pan, F., Wang, B., Ren, D., Hu, X., Perrizo, W.: Proximal support vector machine for spatial data using peano trees: In: ISCA Computer Applications in Industry and Engineering, Las Vegas, Nevada, 2003, pp. 292–297
11. Perrizo, W., Jockheck, W., Perera, A., Ren, D., Wu, W., Zhang, Y.: Multimedia data mining using P-trees. In: International Workshop on Multimedia Data Mining (MDM/KDD), Edmonton, Alberta Canada, 2002, pp. 19–29
12. Rao K.R., Yip, P.: Discrete Cosine Transform: Algorithms, Advantages, Applications. Academic Press Professional, Inc., San Diego (1990)
13. Ruoming J., Agrawal, G.: Efficient decision tree construction on streaming data. In: ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD), pp. 571–576 (2003)
14. Versavel, J.: Road safety through video detection. In: Proceedings of IEEE International Conference on Intelligent Transportation System, Boulder, Colorado, 1999, pp. 753–757
15. Wang, B., Pan, F., Ren, D., Cui, Y., Ding, Q., Perrizo, W.: Efficient OLAP operations for spatial data using peano trees: In ACM Special Interest Group on Management of Data Workshop (SIGMOD), San Diego, California, 2003, pp. 28–34
16. Yao, Y., Gehrke, J.E.: The cougar approach to in-network query processing in sensor networks: ACM Special Interest Group on Data Management of Data (SIGMOD) Record, pp. 9–18 (2002)
17. Zhao, J., Lu, C.T., Kou, Y.: Detecting region outliers in meteorological data. In: Proceedings of ACM International Symposium on Advances in Geographic Information System, New Orleans, Louisiana, 2003, pp.49–55

Streams

► Queries in Spatio-temporal Databases, Time Parameterized

Summary Information

► Metadata and Interoperability, Geospatial

Supplementary Material

► Metadata and Interoperability, Geospatial

Surveillance

► Data Collection, Reliable Real-Time
► Evolution of Earth Observation

Survey Knowledge

► Wayfinding, Landmarks

Susceptibility Analysis

► Sensitivity Analysis

SVG

► Scalable Vector Graphics (SVG)
► Web Mapping and Web Cartography

Sweep Line Algorithm

► Plane Sweep Algorithm

Synchronization of Spatial Data

► Positional Accuracy Improvement (PAI)

Synonymy

► Retrieval Algorithms, Spatial

Taylor Series

► Uncertain Environmental Variables in GIS

Technological Inflection Points in GIS and CAD Development

► Computer Environments for GIS and CAD

Temporal

► Uncertainty, Modeling with Spatial and Temporal

Temporal Data

► Indexing and Mining Time Series Data

Temporal GIS and Applications

MAY YUAN
Department of Geography and Center for Spatial Analysis, University of Oklahoma, Norman, OK, USA

Synonyms

Spatio-temporal information systems; Spatio-temporal informatics; Process; Snapshots; Matrices, geographic; Interaction, space-time; Timestamps; Event; Movement; Activity; Ontology, spatio-temporal; SNAP; SPAN; Reasoning, spatio-temporal

Definition

Geographic information is inherently spatial and temporal. Geographic applications often demand an integrative approach to examine changes and interactions over space and time. Temporal geographic information systems (GIS) are defined here as GIS capable of incorporating temporality into geospatial databases and enabling spatiotemporal query, analysis, and modeling. Adding time into geospatial databases is a far from trivial task. Currently, commercial or public-domain temporal GIS support cell-based spatial data (i. e., rasters) or spatial data objects with one or restricted sets of simple geometries, mostly point- or line-based data only. Research-grade temporal GIS remain limited to pilot studies or prototypes. While comprehensive temporal GIS are still unavailable, much progress has been made to advance the conceptualization, representation, and reasoning of changes, events, processes, and dynamics in geographic worlds. Once implemented, comprehensive, robust temporal GIS can significantly empower an integrative spatial and temporal understanding of the geographic world.

Historical Background

As early as 1961, the geographer James M. Blaut promoted the idea of space as the basic organization of geographic concepts and that *every empirical concept of space must be reducible by a chain of definitions to a concept of process*. His argument represents a classic space-centered view of geography and a process-based approach to uniting space and time. The thought continues in geographic data handling and analysis and in GIS developments. Geographer Brian Berry later proposed geographic matrices of places and characteristics as a basic framework to facilitate place-to-place comparison of geographic variables. He furthermore organized geographic matrices in a temporal sequence to aid the study of changes in spatial association and areal differentiation.

The structure of geographic matrices closely resembles the snapshot approach to incorporating the temporal dimension into GIS databases by time sequencing GIS data layers. Nevertheless, there are also significant differences between the geographic matrices and GIS data layers. Instead of places, GIS data layers use spatial data objects (cells or geometric objects) to represent spatial dimensionality and location and to position spatial data objects in different rows and their characteristics in columns. The strong relational database component in GIS technology further strengthens the space-centered view since a column in a GIS attribute table is a descriptor of a spatial data object; the existence of attributes depends strongly upon the records of spatial objects. If spatial data objects do not exist or have not yet been identified, no geographic characteristics can be described.

On the other hand, a spatial data object may exist without any information about its attributes. In this case, null or zero values are commonly assigned as appropriate to the columns of the spatial data object. Contrarily, switching characteristics (or attributes) from columns to rows allows characteristics to be recorded without explicit association to any predefined space. In other words, characteristics or attributes can exist independently of spatial objects in a geographic table, a distinctive concept from the conventional GIS database design. In both approaches, space and characteristics nevertheless determine the first-level basic geographic data structure, and time is treated as the secondary structure that orders geographic matrices (or spatial attribute tables). Fundamentally, both approaches follow the map paradigm. Information about geographic processes and dynamics (especially movements and flows) cannot be fully captured on maps; neither can processes and dynamics be represented by stages of temporally sequenced geographic matrices or snapshots of GIS data layers.

Uniting space and time is essential to understanding geographic worlds as space–time interactions drive change and movement in geography. Since the late 1980s, many GIScience researchers have proposed numerous approaches to integrating the spatial and temporal dimensions of geographic data. Langran's book entitled *Time in Geographic Information Systems* summarizes the early developments in spatiotemporal database management systems, in which significant efforts focused on historical cadastral mapping [1]. In 2002, Christakos and his colleagues published a book entitled *Temporal GIS: Advanced Functions for Field-Based Applications* [2]. The book introduces a temporal GIS with strong geostatistical and computational support to spatiotemporal mapping and modeling. While the temporal GIS can only handle raster (cell-based) data, the book is a milestone in the development of temporal GIS tools to analyze and model spatiotemporal data. While progress has been made in temporal GIS, conceptual and technical developments as well as many case studies, there is no large-scale robust implementation of these concepts for a comprehensive temporal GIS.

Scientific Fundamentals

There are numerous sources that document the scientific fundamentals and issues in temporal GIS [1,3,4,5]. Many spatiotemporal data models have been proposed covering a wide range of perspectives on spatiotemporal thinking and space–time integration. Many fundamental differences between time and space have been recognized, such as dimensionality (two-dimensional space and one-dimensional time) and directionality (unlimited in space, but one direction in time). Time has been characterized by different types (such as world time, transaction time, and database time), different topologies (such as linear time, branching time, parallel time, and cyclic time), different concepts (such as absolute time, relative time), different reasoning strategies (such as instant-based, interval-based, event-based), and other distinct properties.

In reviewing temporal GIS development and the developments in temporal database systems, Yuan [6] summarized the comparable nature of the two related disciplines. In her analysis, temporal database systems mainly apply time-stamping techniques to incorporate time into relational or object-oriented databases by time-stamping tables, attributes, or values. In parallel, temporal GIS adopts the snapshot model by time-stamping individual data layers, the space–time composite model by time-stamping individual attributes associated with a spatial object, and the space–time object model by time-stamping individual values with attributes of a spatial object.

Spatial reconfiguration is an added complexity in temporal GIS databases that may have disadvantaged the space–time composite model and space–time object model. Because a given attribute can only have one value at a given time for a specified spatial object (i. e., spatial objects are uniform), spatial reconfiguration is necessary whenever changes occur to only a portion of the spatial object. The space–time composite model must restructure spatial objects and reassign spatial identifiers in the new set of spatial objects. Keeping temporal lineages of spatial objects before and after restructuring is not a trivial task, especially when the study area becomes more and more fragmented over time. The space–time object model is able to overcome the issues related to spatial reconfiguration and is useful for recording cadastral histories, for example, but it cannot represent movement, dispersion, and other like geographic processes.

Many event-, process-, movement-, and activity-based temporal GIS data models have been proposed in response to the weaknesses of time-stamping GIS approaches [7,8]. In general, these temporal GIS data models attempt to capture the integrative spatial and temporal dimensions of geographic data and seek the fundamental space–time units meaningful to the chosen application domains. While these approaches have introduced exciting concepts and innovative ideas to integrate spatial and temporal data, the lack of common definitions in terminology and coherent theoretical frameworks presents many challenges to further developments in temporal GIS. For example, the term *event* may refer to a change in data (such as a cell value), an act of changing data values, or a geographic event. As these different approaches represent space–time reality from different perspectives, a common theoretical framework is needed to integrate all the approaches and provide a means to transform data representations among them. The importance of developing common vocabularies and theoretical frameworks cannot be overstated. Some attempts at development have been made through research efforts in hydrological data modeling. However, much work is still needed to develop general theories for spatiotemporal representation, data handling, analysis, and modeling.

Key Applications

As geographic worlds are functions and outcomes of space–time interactions, temporal GIS applications are wide and diverse. From the finest geographic scale to the largest, examples include robot navigation in a room, inventory and object tracking in a warehouse, changes in land ownership and land use, urban growth and sprawl, trip navigation and routing, delivery planning, traffic analysis, analysis of individual daily paths and activities, crime analysis and tracking, forensic investigation, emergency preparedness, rescue operations, globalization and economic development, cultural/political changes and evolution, weather development, spread of wildfires or diseases, pollution dispersion, environmental modeling and management, ecological interactions and processes, climate change, and many, many other research questions and applications that temporal GIS can empower by integrating data and processes in space and time to obtain spatiotemporal understanding of the chosen issues.

Future Directions

A robust temporal GIS has great potential to enable spatiotemporal understanding of geographic worlds and advance the scientific discovery of geographic knowledge. Many challenges remain before this potential can be realized, though. In addition to common vocabularies and theoretical frameworks, research is needed in spatiotemporal ontology, representation, reasoning, query analysis, analytical methods, modeling, and visualization.

On *spatiotemporal ontology*, SNAP and SPAN[1] present upper-level ontologies to distill concepts about space–time occurrents and space–time continuants. Research efforts are needed to expand these and other ontologies relevant to spatiotemporal databases [4,9] to include geographic dynamics, processes, events, and activities for a comprehensive coverage of all kinds of spatiotemporal phenomena and entities. In addition to improving our understanding of space, time, and their interactions, ontologies have been shown to be an effective means for data integration and shall be useful for integrating data across space and time.

Spatiotemporal representation continues to be a challenge because there is not yet a robust framework for analysis and understanding of the intricate and complex properties of space–time in geography. Several researchers have begun coupling the conventional field- and object-based approaches to representations of geographic space with attempts to capture the geographic complexity that often results from space–time interactions. Research on developing a general theory of geographic representation that provides an integrated and holistic view of space, time, fields, and objects is at the heart of temporal GIS development. Some work has been started [10], and further research is urgently needed.

Spatiotemporal reasoning and query analysis has a solid foundation in research from both GIScience and computer science [3,6,11,12,13]. However, complex reasoning and query analysis, such as seeking relationships and effects of events or processes across space and time in a discontinuous manner (e. g., teleconnections of climate events), has not been fully addressed. Methods to elicit environmental correlates and suggest drivers and pathways with which geographic dynamics may thrive or be suppressed are particularly useful in hypothesis building, which probes scientific investigation into new confirmatory findings.

Spatiotemporal reasoning and query analysis can be considered as a precursor to *analysis, modeling, and visualization*. Analytical methods can be used to reveal and confirm causal spatiotemporal relationships. Most spatiotemporal analysis studies emphasize point-based pattern analysis or accessibility/mobility analysis [14], or field-based analysis [2]. Robust methods are needed to analyze linear or areal objects over space and time, in which these objects may change location, shape, and topology. Furthermore, changes to linear and areal objects may incur rotation, merger, split, and clustering. Analytical methods to examine these complex changes and probe potential causal factors to these changes have not yet been fully developed.

Models for inferential statistical testing need to be established to facilitate understanding of spatiotemporal distribution patterns for linear and areal objects. Likewise, spatiotemporal modeling and visualization is under development. Nevertheless, with the growing popularity of research in computational methods, data mining and

[1]The SNAP ontology is consistent with the map metaphor that addresses reality through a series of snapshots. Hence, geographic entities are occurrents at a given point in time. On the other hand, the SPAN ontology applies the idea of 4-dimensionalism in which geographic entities are defined by their volumes in 3D space and 1D temporal occupancy as space-time continuants.

T

knowledge discovery in databases, research in spatiotemporal modeling and visualization has attracted much attention in GIScience. Computational models in geography, traditionally Monte Carlo simulation and Markov chain, have been rapidly expanded in recent years to cellular automata, agent-based modeling, and evolutionary algorithms [15].

In summary, research in temporal GIS and applications has grown over the last two decades. Many fundamental concepts about time and time properties have been discussed in depth in both GIScience and computer science literature. Significant progress has been made in conceptual development as well as in case studies in different application domains. Research emphases have shifted from the space-centered map paradigm to ways of integrating space and time for a richer representation of geographic dynamics. The many challenges and opportunities ahead call for improved communication, more robust theoretical and reasoning frameworks, and more powerful and friendly methods for information query, analysis, modeling and visualization.

Cross References

► Geographic Dynamics, Visualization And Modeling

Recommended Reading

1. Langran, G.: Time in geographic information systems. Taylor and Francis, London (1992)
2. Christakos, G., Bogaert, P., Serre, M.: Temporal GIS: Advanced Functions for Field-Based Applications. Springer, New York (2002)
3. Egenhofer, M.J., Golledge, R.G. (eds.): Spatial and Temporal Reasoning in Geographic Information Systems. Oxford University Press, New York (1998)
4. Frank, A.U.: Ontology for spatio-temporal databases. In: Koubarakis, M., Sellis, T.K., Frank, A.U. (eds.) Spatiotemporal Databases: the Chorochronos Approach, pp. 9–77. Springer-Verlag, Berlin (2003)
5. O'Sullivan, D.: Geogrpahical information science: time changes everything. Prog. Hum. Geogr. **29**, 749–756 (2005)
6. Yuan, M.: Use of a three-domain repesentation to enhance GIS support for complex spatiotemporal queries. Trans. GIS **3**, 137–159 (1999)
7. Peuquet, D.J., Duan, N.: An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. Int. J. Geogr. Inf. Syst. **9**(7) (1995)
8. Worboys, M.F.: Event-oriented approaches to geographic phenomena. Int. J. Geogr. Inf. Sci. **19**, 1–28 (2005)
9. Mark, D.M., Egenhofer, M.J.: Geospatial lifelines. In: Gunther, O., Sellis, T., Theodoulidis, B. (eds.). Integrating Spatial and Temporal Databases. Dagstuhl-Seminar-Report No. 228, http://timelab.co.umist.ac.uk/events/dag98/report.html, Schloss Dagstuhl, Germany (1998)
10. Goodchild, M.F., Yuan, M., Cova, T.J.: Towards a general theory of geographic representation in GIS. Int. J.~Geogr. Inf. Sci. **21**, 239–260 (2007)
11. Frank, A., Grumbach, S., Guting, R.H., Jensen, C.S., Koubarakis, M., Lorentzos, N., Manolopoulos, Y., Nardelli, E., Pernici, B., H.-J. Schek, Scholl, M., Sellis, T., Theodoulidis, B., Widmayer, P.: Chorochronos: a research network for spatiotemporal database systems. SIGMOD Rec. **28**, 12–21 (1999)
12. Hornsby, K., Egenhofer, M.J.: Modeling moving objects over multiple granularities. Ann. Math. Artificial Intell. **36**, 177–194 (2002)
13. Yuan, M., McIntosh, J.: A typology of spatiotemporal information queries. In: Shaw, K., Ladner, R., Abdelguerfi, M. (eds.) Mining Spatiotemporal Information Systems. Kluwer, Norwell, Massachusetts, USA (2002)
14. Miller, H.: Modeling accessiblity using space-time prism concepts within a GIS. Int. J.~Geogr. Inf. Syst. **5**, 287–301 (1991)
15. Torrens, P.M.: Simulating sprawl. Ann. Assoc. Am. Geogr. **96**, 248–275 (2006)

Temporal Indexing

► Indexing and Mining Time Series Data

Temporal Ordered Space Matrix

► Exploratory Visualization

Temporal Resolution

JÉRÔME THÉAU
GIS Training and Research Center,
Idaho State University, Pocatello, ID, USA

Synonyms

Revisit period

Definition

Temporal resolution is defined as the amount of time needed to revisit and acquire data for the exact same location. When applied to remote sensing, this amount of time depends on the orbital characteristics of the sensor platform as well as sensor characteristics. The temporal resolution is high when the revisiting delay is low and vice-versa. Temporal resolution is usually expressed in days.

Main Text

The temporal resolution of a satellite sensor may vary from hours to days. It depends on if the platform orbit is geostationary or not. Moreover, a sun-synchronous orbit follows sun illumination and allows image acquisition at the same time of the day for a location. This characteristic is particularly important for visible-infrared sensors as it makes every image usable (i. e. avoiding image acquisitions during the night) and it, therefore, maximizes the temporal resolution of the sensor. In the case of airborne platforms, the temporal resolution is more variable and can range from days to years, depending on mission planning.

Usually temporal resolution is highly dependent on the spatial resolution of the sensor. The higher the spatial resolution, the lower the temporal resolution is. However, geostationary platforms and pointable sensors are exceptions to this trend.

Here are the temporal resolutions for several common remote sensing products:

- NOAA AVHRR: < 1 day
- MODIS: 1–2 days
- QuickBird: 1–3.5 days (off-nadir)
- Ikonos: 16 days (1.5–3 days off-nadir)
- Landsat ETM+ : 16 days
- RADARSAT1: 24 days (1–6 days off-nadir)
- SPOT5: 26 days (2–3 days off-nadir)
- NAPP (USGS): 5 years

Cross References

► Change Detection
► Co-location Pattern Discovery
► Correlation Queries in Spatial Time Series Data

TerraShare

► Intergraph: Real Time Operational Geospatial Applications

Text Search

► Internet-Based Spatial Information Retrieval

Theoretical Analysis

► Geosensor Networks, Formal Foundations

Theory of Random Functions

► Spatial Uncertainty in Medical Geography: A Geostatistical Perspective

Thiessen Polygons

► Voronoi Diagram

Threshold Accepting Method

► Routing Vehicles, Algorithms

Time Geography

HARVEY J. MILLER
Department of Geography, University of Utah,
Salt Lake City, UT, USA

Synonyms

Activity analysis; Activity theory; Autonomy, space time; Constraints, capability; Constraints, coupling; Constraints, authority; Anchors, space-time; Prism, space-time; Prism, network time; Path, space-time; Location-aware; Regional sience; Activities, fixed; Activities flexible

Definition

Time geography is an individualistic, bottom-up approach to analyzing and simulating human phenomena such as transportation, urban and socio-economic systems. Time geography examines how humans allocate scarce time resources among activities in geographic space, the use of transportation and communication technologies to facilitate this allocation, and the patterns and relationships that emerge from these allocations across the population.

Although researchers and practitioners have utilized the time geographic perspective for over 30 years, it was limited by problems in collecting, storing and analyzing the detailed spatio-temporal data required. Technological and scientific developments have created a major renaissance in time geography since the early 1990s. The development of location-aware technologies (LATs) such as the global positioning system (GPS) and radiofrequency identification (RFID), spatio-temporal and mobile objects databases, and geographic information systems (GIS) has substantially enhanced the ability to collect, store, analyze and communicate detailed space-time activity data and results from analyses and simulations. Time geography also corresponds with an increasing recognition of many human phenomena as complex adaptive systems with emergent properties, and the use of automata for simulating these systems.

Historical Background

Swedish geographer Torsten Hägerstrand developed time geography in the 1960s and 1970s as a regional policy and planning tool [12]. Colleagues and students at the University of Lund in Sweden enhanced and applied time geog-

T

raphy, including Bo Lenntorp's pioneering microsimulations [8]. Time geography was little known outside Sweden until the publication of Hägerstrand's now famous presidential address to the Ninth European Congress of the Regional Science Association, "What about people in regional science?" [3]. A lucid summary and discussion by Allen Pred also helped to popularize time geography outside of Sweden [14]. Lawrence Burns made the first attempt at an analytical foundation for time geography [2].

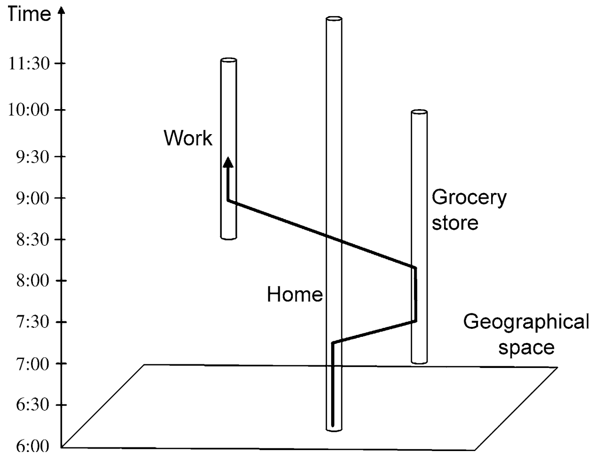
Scientific Fundamentals

Classical Time Geography

Time geography is a constraints-oriented approach to understanding human phenomena. Time geographers as less concerned with explaining individual choice than understanding the constraints on these behaviors imposed by individual activity schedules and the varying resources available to overcome these constraints through trading time for space in movement or communication. Time geography addresses a simple but profound question: "how does participating in an activity at a given place and time limit a person's abilities to participate in activities at other places and times?" [3,14].

Time geography recognizes three major classes of constraints on a person's space-time autonomy: *Capability constraints* limit the activities of individuals through their own physical capabilities and/or the resources they can command. For example, individuals with private automobiles can generally travel faster through space than individuals who walk or rely on public transportation. *Coupling constraints* define where, when and for how long an individual has to join with other individuals for shared activities. *Authority constraints* impose fiat restrictions over particular space-time domains. For example, a private shopping mall can impose more constraints than a traditional city center on individuals' space-time autonomy since private space can be more effectively restricted from occupancy during certain hours and days and for specified purposes.

Capability, coupling and authority constraints combine with the spacing, timing and flexibility of activities to condition space-time autonomy. *Fixed* activities are those that cannot be easily rescheduled or relocated; examples include scheduled work and meetings. *Flexible* activities can be more easily rescheduled and/or can occur at more than one location; these include shopping, recreation, and socializing. These categories can be arbitrary for some activities; however, they provide a powerful mechanism for understanding how the location and timing of some activities such as home and work condition accessibility to other activities. Fixed activities act as *space-time anchors* since

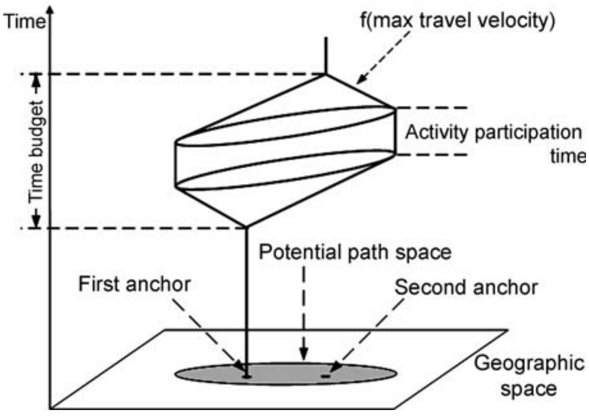


Time Geography, Figure 1 A space-time path

other activities such as shopping and recreation must occur at the temporal gaps between fixed activities. Also, an individual often must use fixed activity locations as bases when trading time for space in movement and communication.

Two central concepts in time geography are the *space-time path* and the *space-time prism*. Figure 1 illustrates a space-time path among activity locations or *stations* in two-dimensional space and time, with time represented by the z-axis orthogonal to the plane. Note that the path is vertical when the individual is stationary at an activity location and becomes more horizontal when she is moving through space. The slope of the path is a function of the apparent movement velocity, i. e., the trading of time for space allowed by the available transportation resources within that environment. Cylinders represent the activity stations in Fig. 1: the length of each cylinder with respect to the z-axis indicating its availability in time.

Figure 2 illustrates a space-time prism. This demarcates the possible locations for the space-time path and is therefore a direct measure of a person's accessibility to the environment and activities. Fixed activities anchor a space-time prism since by definition these allow only one spatial possibility during their duration. For example, the first anchor in Fig. 2 could be the person's home while the second anchor could be their workplace. At some time during the time interval between when the home activity ends and the work activity begins, the person wishes stop at some location to conduct an activity, e. g., shop at a store. The available time interval is the *time budget* for the travel and activity episode. Given these anchoring activities, the time budget, a maximum velocity of movement and the required activity participation time, the prism determines the locations in space and time that are available to the person. The region inside the prism comprises locations where an indi-



Time Geography, Figure 2 A space-time prism

vidual could be at different times during that episode. An activity is not accessible to this person during this episode unless its location and duration intersects with the prism to a sufficient degree, with this determined by the minimum time required to conduct the activity. Similarly, two people cannot meet unless their prisms intersect to a sufficient degree. The projection of the prism to the two-dimensional plane is the *potential path area*: this comprises the region in geographic space where the person can be during the entire time interval.

Geo-spatial Technologies and the New Time Geography

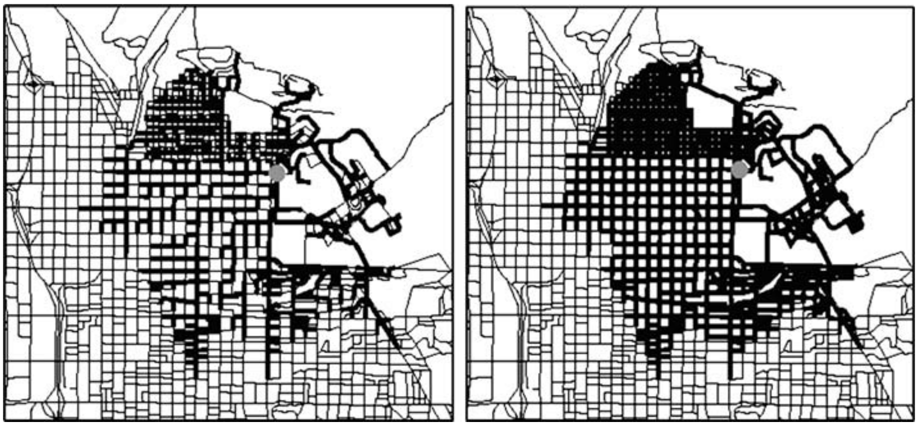
Although classical time geography is an elegant perspective, it has three major weaknesses. First, it assumes a uniform travel velocity across space and time. This is at odds with daily experience in many urban areas where travel velocities vary substantially by location and time of day. Second, classical time geography is only conceptual: basic entities and relationships do not have a rigorous, analytical foundation. This limits its ability to support high-resolution measurements using LATs or spatio-temporal queries in location-based services (LBS) and related applications. Finally, although classical time geography recognizes the potential for virtual interaction, this is greatly subdued relative to the role of physical movement in accessing activities. The time geographic renaissance since the 1990s is addressing these weaknesses using contemporary developments in GIS and related geo-spatial technologies.

Network Time Prisms An elegant method for relaxing the strict assumption of a uniform travel velocity across space is to define the space-time prism using a transportation network with travel time or velocity-attributed network arcs. The ability of GIS to handle detailed georeferenced networks, variable-length attribute segmentation within network arcs, and address-based geolocation has made this strategy attractive.

Figure 3 illustrates two *network time prism* (NTP) products. The left half of the figure illustrates the *potential path tree* (PPT): this determines all nodes within the network that are accessible to the individual given the space-time anchors, network travel velocities and time budget. The right half illustrates the *potential network area* (PNA): this shows all locations in the network (nodes or within arcs) that are accessible to the individual given the same parameters. The PPT is easy to compute, requiring only shortest path tree calculations based on the anchors, but has unrealistic gaps in its network coverage. The PNA is more realistic but requires greater computational effort since it involves constructing a new network topology based on shortest paths from all network nodes [9].

The NTP products illustrated above correspond to a static, single-mode network. It is also possible to derive similar products for dynamic networks with time-varying velocities (e. g., due to flow congestion) and for multi-modal

T



Time Geography, Figure 3 A network potential path tree and potential path area networks (e. g., combined walking and public transit); see [13,17].

Time Geographic Analytical Theory Analytical definitions of the overall space-time path and prism are difficult to derive. Explicit definitions that can support high-resolution measurement and error propagation assessment are possible by temporally disaggregating these entities and solving for the path location or prism footprint at a given instant in time. For example, the spatial extent the space-time prism at any given instant in time t is the intersection of simple and compact spatial sets. At any instant t, the space-time prism defined by anchors $\mathbf{x}_i, \mathbf{x}_j \in \Re^n$ with required presence at times t_i, t_j (respectively) and maximum travel velocity v is:

$$Z_{ij}(t) = \left\{ \mathbf{x} \,\middle|\, f_i(t) \cap p_j(t) \cap g_{ij} \right\} \tag{1}$$

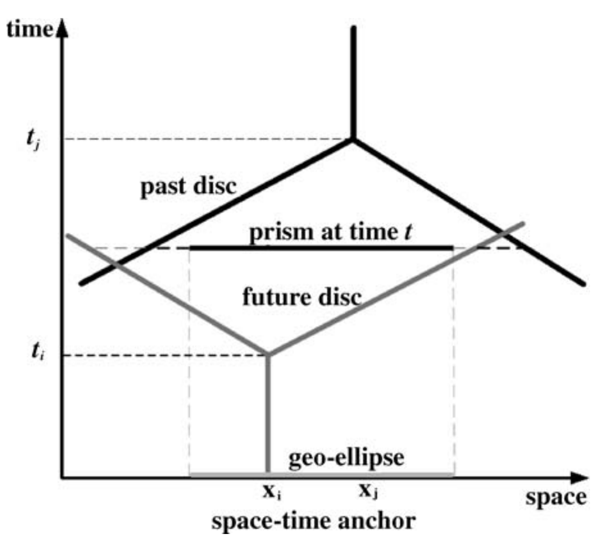
where:

$$f_i(t) = \left\{ \mathbf{x} \,\middle|\, \|\mathbf{x} - \mathbf{x}_i\| \leqslant (t - t_i)\, v \right\} \tag{2}$$

$$p_j(t) = \left\{ \mathbf{x} \,\middle|\, \|\mathbf{x}_j - \mathbf{x}\| \leqslant \left(t_j - t\right) v \right\} \tag{3}$$

$$g_{ij} = \left\{ \mathbf{x} \,\middle|\, \|\mathbf{x} - \mathbf{x}_i\| + \|\mathbf{x}_j - \mathbf{x}\| \leqslant \left(t_j - t_i - a\right) v \right\} \tag{4}$$

$f_i(t)$ is the *future disc*: the locations that can be reached by time t when leaving from $\mathbf{x}_i$ at time t_i $p_j(t)$ is the *past disc*: these are the locations that can reach $\mathbf{x}_j$ by the remaining time $t_j - t$. These are "discs" since they are compact spatial sets consisting of all locations within a fixed distance of a point. g_{ij} is the *geo-ellipse*: it constrains the prism locations to account for any stationary activity time a during the time interval. It is equivalent to the *potential path area* of classical time geography. This is an ellipse since consists of all locations within a fixed distance of two locations. Figure 4 provides a conceptual illustration [10]. The sets defined in Eqs. (1)–(4) are simple geometric forms: the future and past discs are lines in one spatial dimension, circles in two dimensions and spheres in three dimensions. The geo-ellipse is a line, ellipse and ellipsoid in one, two and three spatial dimensions respectively. We can solve for these sets and their intersections analytically or (in some cases) tractable and accurate numerical methods. It is also possible to simplify the calculations by solving for time boundaries to determine temporal subintervals when a set is encompassed by other sets and can be ignored [10].



Time Geography, Figure 4 Analytical definition of the space-time prism

Time Geography and Virtual Interaction From a time geographic perspective, we can characterize communication modes based on their spatial and temporal constraints. Communication modes can require physical presence or allow telepresence. Similar, communication can involve synchronous or asynchronous interaction. This leads to four types of communication modes; see Table 1. *Synchronous presence* (SP) corresponds to face-to-face (F2F) interaction. *Synchronous telepresence* (ST) requires only coincidence in time; this includes telephones, text messaging and television. *Asynchronous presence* (AP) requires coincidence in space but not time; this includes a written note pasted on an office door. *Asynchronous telepresence* (AT) does not require coincidence in space and time: this mode includes printed media, email, text messages and webpages [5].

Time Geography, Table 1 Spatial and temporal constraints on communication

	Spatial	
Temporal	**Physical presence**	**Telepresence**
Synchronous	**SP**	**ST**
	Face to face (F2F)	Telephone
		Instant messaging
		Television
		Radio
		Teleconferencing
Asynchronous	**AP**	**AT**
	Written notes	Mail
	Hospital charts	Email
		Fax machines
		Printed media
		Webpages

Conceptual extensions of the space-time path and prism can accommodate the communication modes in Table 1. Current GIS data models and tools such as variable length arc segmentation can implement this extended framework [18]. It is also possible to rigorously derive the spatial and temporal constraints on interaction implied by the communication modes in Table 1 by introducing the concept of *message windows* (intervals of time corresponding to potential or actual communication events) and solve for the temporal constraints on message windows using the well-known Allen time predicates that encompass all possible relationships between two intervals of time [11].

Key Applications

Mobile Objects Databases

Time geography provides an elegant framework for representing mobile objects over different levels of temporal granularity. This can support the development of spatio-temporal queries from mobile objects databases [4].

Location-Based Services

First-generation LBS can answer simple queries regarding the availability of services and navigation for an immediate task. They are less successful at more sophisticated queries involved in the sequencing and timing of tasks and subtasks over an extended period of time. Time geography can provide the foundation for next-generation LBS that account not only for the immediate task at hand but also its context with respect to the individual's required and desired activities over a given time frame [15].

Transportation Analysis

Traditional transportation models are aggregate and "top-down," specifying a set of linked or simultaneous equations, or a constrained optimization problem, that treats travel demands similar to water flowing in pipes. This ignores individual characteristics critical to transportation behavior, as well as emergent properties of transportation systems. Individualistic *activity-based analysis* methods based on time geographic principles can capture individual characteristics and emergent transportation properties; see [16].

Urban Simulation

Traditional urban modeling also follows top-down, aggregate strategies with simple dynamics, unrealistic market mechanisms, and archaic spatial structures such as monocentric and polycentric urban foci. Bottom-up approaches based on time geography and using methods such as cellular automata and agent-based modeling can capture realistic behaviors, complex dynamics and do not pre-suppose simple urban spatial structures for tractability; see [1].

Public Health

Most epidemiological models assume a simple mixing mechanism where all individuals in a community are equally likely to interact and therefore spread an infectious disease. Time geography supports a more realistic view of interpersonal contact and can therefore improve the explanatory and predictive power of disease propagation models; see [6]. Similarly, time geography can support a more comprehensive assessment of a person's exposure to environmental risk factors over time frames ranging from a day to a lifetime.

Future Directions

Research frontiers in time geography include the following critical issues.

Visualization and Knowledge Discovery

Current geo-visualization and geographic knowledge discovery applications in time geography are limited to modest data sets of a few hundred paths or prisms. However, the potential size of space-time activity datasets generated from LATs and LBS approaches thousands, perhaps millions, of records. Some progress is being made in the mobile objects database literature; these methods need to be extended to encompass the broader range of concepts in time geography; see [7].

Scalable Techniques

A related issue is a need to develop and apply scalable techniques to massive space-time activity databases. Time geographic analysis can be decomposed into parallel computations based on task and/or data parallelism; computational techniques such as those based on parallel or grid computing architectures have promise to reveal new knowledge from these databases.

Multiscale Linkages

Time geography has conceptual linkages from the individual to aggregate entities and relationships such as bundling of space-time paths due to coupling constraints or the emergence of complex space-time systems. However, there is still a need for analytical and modeling linkages. Of practical importance is a need for synoptic measures of patterns and dynamics from individual-level data and simulations.

Locational Privacy

The possibility of unwarranted surveillance and control through LATs and LBS is considerable. Required are techniques and protocols based on the evolving concept of location privacy to preserve the scientific value of time geographic data without violating individual privacy and freedom.

Cross References

▶ Dynamic Travel Time Maps
▶ Temporal GIS and Applications

Recommended Reading

1. Benenson, I., Torrens, P. M.: Geosimulation: Automata-Based Modeling of Urban Phenomena. Wiley (2004)
2. Burns, L.D.: Transportation, Temporal and Spatial Components of Accessibility. Lexington Books (1979)
3. Hägerstrand, T.: What about people in regional science? Papers Reg. Sci. Association **24**, 7–21 (1970)
4. Hornsby, K., Egenhofer, M. J.: Modeling moving objects over multiple granularities. Ann. Math. Artif. Intell. **36**, 177–194 (2002)
5. Janelle, D.G.: Impact of information technologies. in Hanson, S., Giuliano, G. (eds.) The Geography of Urban Transportation, 3ed.. Guilford, 86–112 (2004)
6. Jacquez, G. M., Goovaerts, P., Rogerson, P.: Design and implementation of a space-time intelligence system for disease surveillance. J. Geogr. Syst. **7**, 7–23 (2005)
7. Laube, P., Imfeld, S., Weibel, R.: Discovering relative motion patterns in groups of moving point objects. Int. J. Geogr. Inf. Sci. **19**, 639–668 (2005)
8. Lenntorp, B.: Paths in space-time environments: A time geographic study of movement possibilities of individuals. Lund Studies in Geography Number 44, Royal University of Lund, Sweden (1976)
9. Miller, H.J.: Measuring space-time accessibility benefits within transportation networks: Basic theory and computational methods. Geogr. Anal. **31**, 187–212 (1999)
10. Milller, H.J.: A measurement theory for time geography. Geogr. Anal. **37**, 17–45 (2005)
11. Miller, H.J.: Necessary space-time conditions for human interaction. Environ. Plann. B: Plann. Des. **32**, 381–401 (2005)
12. Öberg, S.: Hägerstrand and the remaking of Sweden. Prog. Hum. Geogr. **29**, 340–349 (2005)
13. O'Sullivan, D., Morrison, A., Shearer, J.: Using desktop GIS for the investigation of accessibility by public transport: An isochrone approach. Int. J. Geogr. Inf. Sci. **14**, 85–104 (2000)
14. Pred, A.: The choreography of existence: Comments on Hagerstrand's time-geography and its usefulness. Econ. Geogr. **53**, 207–221 (1977)
15. Raubal, M., Miller, H.J., Bridwell, S.: User centered time geography for location-based services. Geogr. Annaler B **86**, 245–265 (2004)
16. Timmermans, H., Arentze, T., Joh, C.-H.: Analyzing space-time behaviour: New approaches to old problems. Prog. Hum. Geogr. **26**, 175–190 (2002)
17. Wu, Y-H. and Miller, H.J.: Computational tools for measuring space-time accessibility within transportation networks with dynamic flow. J. Transp. Stat. **4**(2/3), 1–14 (2002)
18. Yu, H.: Spatio-temporal GIS design for exploring interactions of human activities. Cartography Geogr. Inf. Sci. **33**, 3–19 (2006)

Time of Flight

▶ Indoor Localization

Time-Series

▶ Indexing, High Dimensional

Time-Series Correlation Graph

▶ Geographic Dynamics, Visualization And Modeling

Timeseries Data

▶ Computing Fitness of Use of Geospatial Datasets

Timestamps

▶ Temporal GIS and Applications

TIN

▶ Photogrammetric Products

Tiny Aggreagation Service

▶ Geosensor Networks, Estimating Continuous Phenonena

Top-k OLAP Queries Using Augmented Spatial Access Methods

Nikos Mamoulis[1], Spiridon Bakiras[1], Panos Kalnis[2]
[1] Department of Computer Science, University of Hong Kong, Hong Kong, China
[2] Department of Computer Science, National University of Singapore, Singapore, Singapore

Synonyms

Top-k OLAP queries using multi-resolution tree structures; OLAP query; Top-k query

Definition

A top-k OLAP query groups measurements with respect to some abstraction level of interesting dimensions and selects the k groups with the highest aggregate value. An example of such a query is "find the 10 combinations of product-type and month with the largest sum of sales". Top-k queries may also be applied in a spatial database context, where objects are augmented with some measures that must be aggregated according to a spatial division. For instance, consider a map of objects (e. g., restaurants) where each object carries some non-spatial measure (e. g., the number of customers served during the last month). Given a partitioning of the space into regions (e. g., city districts), the goal is to find the regions with the highest number of served customers. Formally, the top-k OLAP query can be defined as follows:

Let $\mathcal{D} = \{d_1, \ldots, d_m\}$ be a set of m interesting dimensions and assume that the domain of each dimension $d_i \in \mathcal{D}$ is partitioned into a set $R_i = \{r_i^1, \ldots, r_i^{|R_i|}\}$ of $|R_i|$ ad-hoc or predefined ranges based on some hierarchy level. Let k be a positive integer. An OLAP top-k query on $\mathcal{D}$ selects the k groups $g_1, \ldots, g_k$ with the largest aggregate results, such that $g_j = \{r_1, \ldots, r_m\}$ and $r_i \in R_i$, $\forall i \in [1, m]$.

An example top-10 OLAP query could be expressed by the SQL statement that follows. Here, the partition ranges at each dimension are implicitly defined by levels of hierarchy (type for products and city for stores).

```
SELECT product-type, store-city,
                     sum(quantity)
FROM Sales
GROUP BY product-type, store-city
ORDER BY sum(quantity)
STOP AFTER 10;
```

Historical Background

Data warehouses integrate and summarize large amounts of historical information accumulated from operational databases. On-line Analytical Processing (OLAP) refers to the set of operations that are applied on a Data Warehouse to assist analysis and decision support. Data warehouses are usually modeled by the star schema [7], where some measures (e. g., sales) are analyzed with respect to some interesting dimensions (e. g., products, stores, time, etc.), representing business perspectives. A *fact* table stores records corresponding to transactions that have been consolidated in the warehouse. One or more columns in the fact table capture the measures, while each remaining attribute stores values for a dimension at the *most refined* abstraction level. For example, a tuple in the fact table stores a transaction for a particular product-id sold at a particular store-id at some particular time instant. A dimensional table models multi-level hierarchies of a particular dimension. For example, a tuple in the dimensional table `product` stores information about the color, type, manufacturer, etc., for each product-id. The star schema was extended in [3] to include spatial abstraction levels and dimensions. The location of stores where products are sold is an example of a spatial attribute, with respect to which of the sales could be analyzed (possibly together with non-spatial attributes of other dimensions). One can also define hierarchies for spatial attributes. In general, hierarchies of spatial and non-spatial ordinal attributes can be defined either by predefined decompositions of the value ranges (e. g., exact location, city, county, state, country, etc.) or by ad-hoc partitioning techniques (e. g., by a regular spatial grid of arbitrary granularity).

Data analysts pose OLAP queries that summarize the fact table information with respect to the interesting dimensions at some particular level of their hierarchies, e. g., "retrieve the total sales per month, product color, and store location". An ideal method to manage a data warehouse, in order to answer OLAP queries efficiently, is to materialize all possible groupings of the measures with respect to every combination of dimensions and hierarchies thereof. In this way, the result of each OLAP query could directly be accessed. Unfortunately, this technique is infeasible because a huge space is required for storing the results for all possible combinations and a long time is required to maintain these combinations after updates in the warehouse. In view of this, several partial materialization techniques [4] select from the complete hierarchy of possible hyper-cubes those that assist the evaluation of most frequent OLAP queries and at the same time they meet the space and maintenance time constraints. Methods for view selection have been extended for spatial data warehouses [3,10], where the spatial dimension plays an important role due to the ad-hoc nature of groups there. Nevertheless, these techniques cannot deal with ad-hoc groupings of the dimensional ranges, which may still have to be evaluated directly on base tables of the data warehouse. This is particularly the case for spatial attributes for which the grouping hierarchies are mostly ad-hoc.

Papadias et al. [9] proposed a methodology that remedies the problem of ad-hoc groupings in spatial data warehouses. Their method is based on the construction of an *aggregate* R–tree [8] (simply aR–tree) for the finest granularity of the OLAP dimensions (i. e., for the fact table data). OLAP group-by queries for ad-hoc groupings of dimensions are answered by spatially *joining* the regions defined by the cube cells with the tree.

An interesting OLAP query generalization is the *iceberg* query [2]; the user is only interested in cells of the cuboid with aggregate values larger than a threshold t (e. g., "find the sum of sales for each combination of product-type and month, only for combinations where the sum of sales is greater than 1000"). The top-k OLAP query [6] is a variant of the iceberg query that groups measures in a cuboid and only returns the k cells with the largest aggregate values (e. g., "find the 10 combinations of product-type and month with the largest sum of sales").

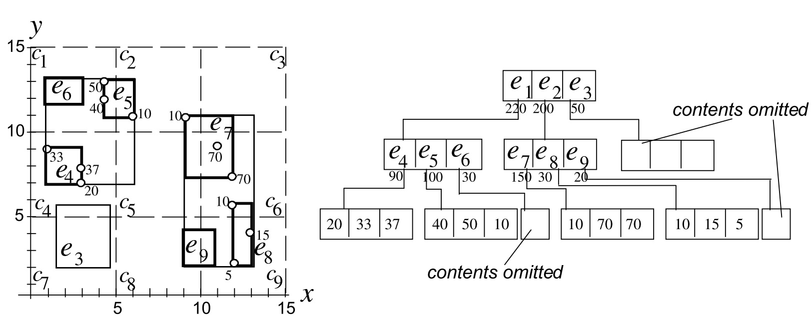
Scientific Fundamentals

A naive way to process top-k OLAP queries (and iceberg queries) is to perform the aggregation for each cell and then select the cells with the highest values. A more efficient method [6] utilizes an *aggregate* R–tree [8,9] (simply aR–tree) in combination with branch-and-bound search heuristics [5] to efficiently compute the query result. The aR–tree is structurally similar to the R*–tree [1], however, it is not used to index object-ids, but *measures* at particular locations (which could be mixtures of spatial co-ordinates and ordinal values of non-spatial dimensions at the finest granularity). The main difference to the R*–tree is that each directory node entry e is augmented with aggregate results for all measures indexed in the sub-tree pointed by e. Figure 1 shows an exemplary aR–tree (the ids of entries at the leaf level and the contents of some nodes are omitted). The value shown under each non-leaf entry e_i corresponds to an aggregate value (e. g., sum) for all measures in the subtree pointed by e_i.

To process a top-k OLAP query, traverse the aR–tree in a branch-and-bound manner, following entries that have the highest probability to contribute to cells of large aggregate results. By detecting these dense cells early, one is able to minimize the number of visited tree nodes until the termination of the algorithm. As an example, assume that there are only two (spatial) dimensions x and y with integer values ranging from 0 to 15. In addition, assume that each dimension is partitioned into value ranges [0,5), [5,10), [10,15]. Figure 1 shows a set of measures in this space indexed by an aR–tree and the 3×3 groups (cells) $c_1, \ldots, c_9$ defined by the combinations of partition ranges. By visiting the leftmost path of the tree, it is clear that the result $c_4.agg$ for cell c_4 is between 90 (due to the contents of e_4) and 140 (due to e_3 that overlaps c_4). Thus, one can set *lower* $c_4.lb$ and upper $c_4.ub$ bounds for the aggregate result in c_4, and accordingly for all cells in space. In addition, based on the information derived by traversing the tree, one can maintain a set *LB* of k cells with the largest lower bounds. Let t be the k-th largest lower bound. Let e_i be an aR–tree entry. If for all cells c that intersect e_i, $c.ub \le t$, then the subtree pointed by e_i cannot contribute to any top-k result, and thus it can be pruned from search. Intuitively, the algorithm terminates after exactly computing the contents of some cells, and non-visited subtrees overlap only with cells that cannot end up in the top-k result.

An order for visiting the nodes of the tree remains to be defined. This order should maximize the pruning power and allow the algorithm to terminate as early as possible. Observe that an entry e_i of a subtree (not visited yet) that intersects a number of cells can contribute at most $e_i.agg$ to the aggregate result of the cell. For example, in Fig. 1, even though the contents of the subtree pointed by e_3 are unknown, it is known that c_4 can contribute at most 50 to this cell. In addition, for an entry e_i which is totally contained in a cell c, it is known that it contributes exactly $e_i.agg$ to c, without having to access the subtree pointed by e_i. For example, visiting the leaf node pointed by e_4 is pointless since the MBR of the entry is totally contained in c_4, therefore it is clear that c_4 gets exactly 90 from this entry. These observations lead to the design of a top-k OLAP algorithm, which is described in Fig. 2.

During the *initialization* phase of the algorithm, one can visit the root node of the aR–tree and compute upper and lower bounds for all cells based on their overlap with root entries (lines 1–9). In the running example, $e_1.agg$, $e_2.agg$, $e_3.agg$ was used to compute $c_1.ub = 220$, $c_2.ub = 420$, $c_3.ub = 200$, etc. In addition, $c_i.lb = 0$ for all cells c_i since no entry is totally contained in one of them. Assuming that $k = 1$ (in this example) and based on the information so far, the algorithm cannot terminate since the highest lower bound is smaller than some upper bound.



Top-k OLAP Queries Using Augmented Spatial Access Methods, Figure 1 Top-k grouping example

```
Algorithm TopkOLAP(aR–tree T, k)
1.  LB := ∅; t := 0; c.lb := c.ub := 0, for all cells;
2.  n := root(R);
3.  for each entry e_i ∈ n do
4.     if e_i.MBR is contained in a cell c then
5.        c.lb := c.lb + e_i.agg; c.ub := c.ub + e_i.agg;
6.        add/update c in LB; /*heap of lower bounds*/
7.        t := k-th largest value in LB;
8.     else /*not contained*/
9.        for each cell c intersected by e_i set c.ub := c.ub + e_i.agg;
10. for each entry e_i ∈ n do
11.    e_i.ub := max{c.ub, ∀ cells c intersected by e_i};
12.       add e_i on a max-heap H; /*organize H primarily by e_i.ub; break ties, using e_i.agg*/
13. while notempty(H) do
14.    e := H.top;
15.    if e.ub ≤ t then break; /*termination condition*/
16.    n := load aR–tree node pointed by e;
17.    C := all cells c intersected by e;
18.    for each cell c ∈ C set c.ub := c.ub − e_i.agg;
19.    for each entry e_i ∈ n do
20.       if e_i.MBR is contained in a cell c then /* always true if n is a leaf node */
21.          c.lb := c.lb + e_i.agg; c.ub := c.ub + e_i.agg;
22.          add/update c in LB;
23.          t := k-th largest value in LB;
24.       else /*not contained*/
25.          for each cell c intersected by e_i set c.ub := c.ub + e_i.agg;
26.    for each entry e_i ∈ n not contained in a cell do
27.       e_i.ub := max{c.ub, ∀ cells c intersected by e_i};
28.       add e_i on H;
29.    for each entry e_j ∈ H overlapping some cell in C do
30.       e_j.ub := max{c.ub, ∀ cells c intersected by e_j};
31.       update e_j's position in H, if necessary;
```

Top-k OLAP Queries Using Augmented Spatial Access Methods, Figure 2 The basic algorithm for top-k OLAP queries

At this stage, it is necessary to determine which node to visit next. Intuitively, an entry which intersects the cell with the greatest upper bound should be followed first in order to decrease this upper bound and at the same time to increase the lower bounds of other cells, potentially leading to an early termination of the algorithm. In addition, from all entries intersecting the cell with the greatest upper bound, the one with the largest *e.agg* should be visited first since it is likely to mostly contribute to the cells it overlaps. Thus, one must *prioritize* the entries to be visited according to the above criteria and follow a *best-first* search order. In other words, all entries (i. e., subtrees) of the aR-tree that have not been followed yet are organized in a heap H (i. e., priority queue). The entry e to be followed next is the one with the greatest *e.agg* from those intersecting the cell with the greatest upper bound. In our example, after visiting the root, e_1,e_2,e_3 are inserted into H and e_1 becomes the top element since it intersects c_2 (and c_5) having $c_2.ub = 420$ and $e_1.agg > e_2.agg$ (e_2 also intersects c_2 and c_5). Lines 10–12 of the algorithm compute the heap order key for the root entries and insert them to H.

When de-heaping an entry e from H, visit the corresponding node n at the aR–tree. Let C be the set of cells intersected by e. The first thing to do is to decrease the upper bounds of cells in C by *e.agg* since these bounds will be refined by the entries of the new node n. For each entry $e_i \in n$, two cases are once again considered; (i) e_i is totally contained in a cell, or (ii) e_i overlaps more than one cells. In the first case, one must only update the lower bound of the covering cell. Otherwise, add $e_i.agg$ to the upper bounds of all cells that intersect e_i. Note that for entries at the leaf level, only case (i) applies. After processing all entries, the upper bounds of all cells in C are updated. Based on these new bounds, the heap key of the newly processed entries are computed (only for case (ii) entries) and added on H. In addition, for entries that are already in H and intersect any cell in C, the positions in H are changed, if necessary, considering the new upper bounds of these cells.

The algorithm terminates (line 15) if for the entry e that is de-heaped $e.ub \leq t$, where t is the smallest of the top-k results found so far (stored in LB). Indeed, if this condition holds, no cell can potentially have a higher aggregate value than the current k-th result.

To exemplify the functionality of the algorithm, consider the data and the tree of Fig. 1 and assume that one must find the cell with the highest aggregate value (i. e., $k = 1$). Start by examining the three root entries. After computing $e_1.ub = 420$, $e_2.ub = 420$, and $e_3.ub = 270$, add them on H. e_1 becomes the top heap element since $e_1.agg > e_2.agg$.

After de-heaping e_1, load the aR–tree node pointed by it. First, reduce the upper bounds of c_1,c_2,c_4,c_5 by $e_1.agg = 220$. Entry e_4 is totally covered by cell c_4, therefore $c_4.lb = 90$. Note that it is not necessary to visit the node pointed by e_4. c_4 now becomes the best result and $t = 90$. Entry e_5 overlaps cells c_1 and c_2, increasing their upper bounds by $e_5.agg$. Finally, e_6 is fully contained in c_1 and sets $c_1.lb = 50$. The upper bounds of e_2 and e_3 are updated to 300 (due to c_2) and 140 (due to c_4), respectively. In addition, e_5 has been added to H with $e_5.ub = 300$ (due to c_2). The next entry to be de-heaped is e_2. Since $e_2.ub > t$, the algorithm does not terminate and one can load the corresponding node and examine its entries which are all added on H. The top heap entry is now e_7 with $e_7.ub = 250$ (due to c_2). Still, $e_7.ub > t$ and the pointed node by it is popped, which is a leaf node. The currently best cell now becomes c_6 with $c_6.lb = 140$. In turn, the top heap entry is e_8, with $e_8.ub = 170$ (due to c_6). After visiting the leaf node pointed by e_8, $c_6.lb$ becomes 150, which is the current t. The algorithm now terminates because the next entry popped from H is e_3 with $e_3.ub = 140 < t$.

Reference [6] discusses extensions of this basic algorithm for the case where the total number of cells is larger than the available memory (this can happen when very refined partitions are present at each dimension). As a final note, the top-k OLAP query evaluation algorithm can also be applied for iceberg queries [2] after replacing the floating bound of the k-th cell by the fixed bound t, expressed in the iceberg query.

Key Applications

Sciences

Top-k OLAP queries can find application in scientific data analysis by Spatial OLAP. They can be used to identify spatial regions that are "heavy hits" with respect to an aggregate measure. For instance, they can be used to identify the districts where people with the highest average income live. Other scientific applications include astronomy (e. g., find constellations with high average brightness, size, or distance to earth) and biology (e. g., find groups of spatially close cells with extreme behavior or reaction to some stimulus).

Decision Support

Top-k OLAP queries can assist decision makers in marketing applications to identify groups of attribute values (e. g., product-ids, sale-regions) of high aggregate measure values (e. g., sales). These results may help for the better design of target marketing (e. g., what are the popular items and why are they popular in some regions more than in some others?).

Data Mining

Many data mining applications are based on frequent itemset counting. Top-k OLAP queries can help identify the groups of items that appear together very frequently in sales transactions.

Information Retrieval

Document clustering in Information Retrieval is based on identifying document pairs having a large set of frequently appearing common terms in them. A top-k OLAP query can be used to rank document pairs based on this measure. Similarly, one can rank pairs of words based on how frequently they appear together in documents.

Future Directions

There is room for future work on top-k OLAP query processing. First, the problem has not received adequate attention for the case where the data to be aggregated are not indexed by aR–trees. In this case, extensions of hash-based and sort-based techniques, like those proposed for aggregate queries in [2], should be developed. Second, current research has only considered groups that correspond to *disjoint* partitions of the high-dimensional aggregated space. It would be interesting to study the problem for overlapping partitions; e. g., what are the top-k 1 km × 1 km spatial regions with the highest aggregate score? This problem is more challenging due to the great increase of groups to be considered.

Cross References

► Abstraction of GeoDatabases
► Aggregation Query, Spatial
► Data Analysis, Spatial
► Indexing, High Dimensional
► OLAP, Spatial

Recommended Reading

1. Beckmann, N., Kriegel, H., Schneider, R., Seeger, B.: The R*–tree: An efficient and robust access method for points and rectangles. In: Proc. of ACM SIGMOD, pp. 220–231 (1990)
2. Fang, M., Shivakumar, N., Garcia-Molina, H., Motwani, R., Ullman, J. D.: Computing iceberg queries efficiently. In: Proc. of VLDB (1998)
3. Han, J., Stefanovic, N., Koperski, K.: Selective materialization: An efficient method for spatial data cube construction. In: Second

Pacific-Asia Conference, PAKDD-98, pp. 144–158, Melbourne, Australia, 15–17 Apr 1998
4. Harinarayan, V., Rajaraman, A., Ullman, J. D.: Implementing data cubes efficiently. In: Proc. of ACM SIGMOD, pp. 205–216 (1996)
5. Hjaltason, G. R., Samet, H.: Distance browsing in spatial databases. ACM TODS **24**(2), 265–318 (1999)
6. Kalnis, P., Mamoulis, N., Bakiras, S.: On discovering moving clusters in spatio-temporal data.In: Proceedings of the 9th International Symposium on Spatial and Temporal Databases (SSTD), pp. 364–381, Angra dos Reis, Brazil, August 2005
7. Kimball, R.: The Data Warehouse Toolkit. John Wiley, USA (1996)
8. Lazaridis, I., Mehrotra, S.: Progressive approximate aggregate queries with a multi-resolution tree structure. In: Proc. of ACM SIGMOD (2001)
9. Papadias, D., Kalnis, P., Zhang, J., Tao, Y.: Efficient OLAP operations in spatial data warehouses. In: Proc. of SSTD (2001)
10. Pedersen, T. B., Tryfona, N.: Pre-aggregation in spatial data warehouses. In: Proc. of SSTD, pp. 460–480 (2001)

Top-K OLAP Queries Using Multi-Resolution Tree Structures

► Top-k OLAP Queries Using Augmented Spatial Access Methods

Top-K Query

► Top-k OLAP Queries Using Augmented Spatial Access Methods

Top-K Query Processing

► Top-k Retrieval Techniques in Distributed Sensor Systems

Top-k Retrieval Techniques in Distributed Sensor Systems

SONG LIN[1], DEMETRIOS ZEINALIPOUR-YAZTI[2], DIMITRIOS GUNOPULOS[1]
[1] Department of Computer Science and Engineering, University of California, Riverside, CA, USA
[2] School of Pure and Applied Sciences, Open University of Cyprus, Nicosia, Cyprus

Synonyms

Top-k query processing; Spatio-temporal similarity search

Definition

Fast developments in wireless technologies and microelectronics made it feasible to develop economically viable embedded sensor systems for monitoring and understanding the physical world [5]. Traditional monitoring approaches, like passive sensing devices, transmit their readings to a centralized processing unit for storage and analysis. *Wireless Sensor Devices (WSDs)* on the other hand, are tiny computers on a chip that is often no bigger than a coin or credit card. These devices, equipped with a low frequency processor ($\approx$ 4–58 MHz) and a wireless radio, can sense parameters such as, light, sound, temperature, humidity, pressure, noise levels, movement, and many others at extremely high resolutions. The applications of sensor networks range from environment monitoring (such as atmosphere and habitant monitoring, seismic and structural monitoring) to industry manufacturing (such as factory and process automation). WSDs are extremely resource constrained devices with a limited energy budget. Energy is usually the primary concern when designing an in-network algorithm for sensor networks.

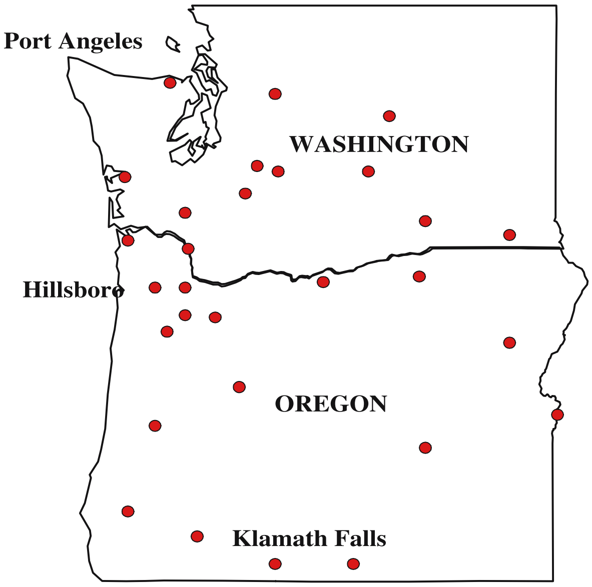
The main function of the sensor network is to measure the environmental quantities and return relevant information that may be interest to the user. In many occasions, it is less expensive and more meaningful to find the k highest ranked (or *Top-K*) answers rather the entire set of answers, if this can minimize the evaluation cost of the query. Since users are usually only interested in a few most relevant answers to their query, the goal of a top-k query is to return manageable result sets consisting of these most relevant answers. Figure 1 shows a sensor network with five sensors $s_1, \dots, s_5$. In Fig. 2 the 5 sensors are measuring the temperatures of some area in five consecutive hours ($o_1, \dots, o_5$). Once a data record is generated by a sensor, it is stored at the sensor locally. A top-k ($k = 1$) query on the sensor data finds the hour during which the average reading of all the 5 sensors is the highest.

Here we propose techniques to overcome the inherent problems of the centralized data acquisition scenario. Specifically, we formulate two novel problems, Distributed Top-k Retrieval and Distributed Top-K Trajectory Retrieval, and solve these problems by proposing a family of efficient threshold algorithms.

Distributed Top-k Retrieval in Sensor Networks

Let R be a relation with n attributes $s_1, s_2, \dots, s_n$, each featuring m objects $o_1, o_2, \dots, o_m$. The jth attribute of the ith object is denoted as o_{ij}. Also let $G(V, E)$ denote an undirected network graph that interconnects the n vertices in V using the edge set E. The edges in E, represent the connections between the vertices in V (the set of sensor nodes).

T



Top-k Retrieval Techniques in Distributed Sensor Systems, Figure 1
The Intel Lab sensor network

s_1 oid,val	s_2 oid,val	s_3 oid,val	s_4 oid,val	s_5 oid,val	Average oid,val
o_1, 63	o_1, 61	o_1, 1	o_1, 28	o_1, 35	o_1, 188/5=38
o_2, 66	o_2, 91	o_2, 92	o_2, 56	o_2, 58	o_2, 363/5=73
o_3, 48	o_3, 1	o_3, 16	o_3, 56	o_3, 54	o_3, 175/5=35
o_4, 99	o_4, 90	o_4, 75	o_4, 74	o_4, 67	o_4, 405/5=81
o_5, 44	o_5, 7	o_5, 70	o_5, 19	o_5, 67	o_5, 207/5=41

Top-k Retrieval Techniques in Distributed Sensor Systems, Figure 2
The local scores of five objects $o_1, \ldots, o_5$ which are located at five different sensors $s_1, \ldots, s_5$. The last column displays the average of the scores (overall rank)

We assume that each vertex is connected to only d ($d \ll n$) other vertices (i. e. the average degree of the graph is d). Now assume that each sensor s_i is mapped to the elements of the vertex set $V = \{v_1, v_2, \ldots, v_n\}$ using a $1:1$ mapping function $f : s_i \to v_i$, $\forall i$. This happens because each sensor maintains only local information (i. e. a single dimension in the n-dimensional space).
Consider $Q = (q_1, q_2, \ldots, q_n)$, a top-k query with n attributes. Each attribute of Q refers to the corresponding attribute of an object and the query attempts to find the k objects which have the maximum value in the following scoring function:

$$\text{Score}(o_i) = \sum_{j=1}^{n} w_j * \text{sim}(q_j, o_{ij}) \quad (1)$$

where $\text{sim}(q_j, o_{ij})$, is a similarity function which evaluates the jth attribute of the query Q against the jth attribute of an object o_i and returns a value in the domain $[0, 1]$ (1 denotes the highest similarity). Since each attribute might have a different factor of importance, we also use a weight factor w_j ($w_j > 0$), which adjusts the significance of each attribute according to the user preferences. For instance, if the readings acquired by node v_l in the network are more important than the readings acquired by the other nodes then w_l might be set to a large value. Note that, similarly to [2,6], we require the score function to be *monotone*. A function is monotone if the following property holds: if $\text{sim}(q_j, o_{1j}) > \text{sim}(q_j, o_{2j})$ ($\forall j \in n$) then $\text{Score}(o_1) > \text{Score}(o_2)$. This is true when $w_j > 0$ ($\forall j \in n$).

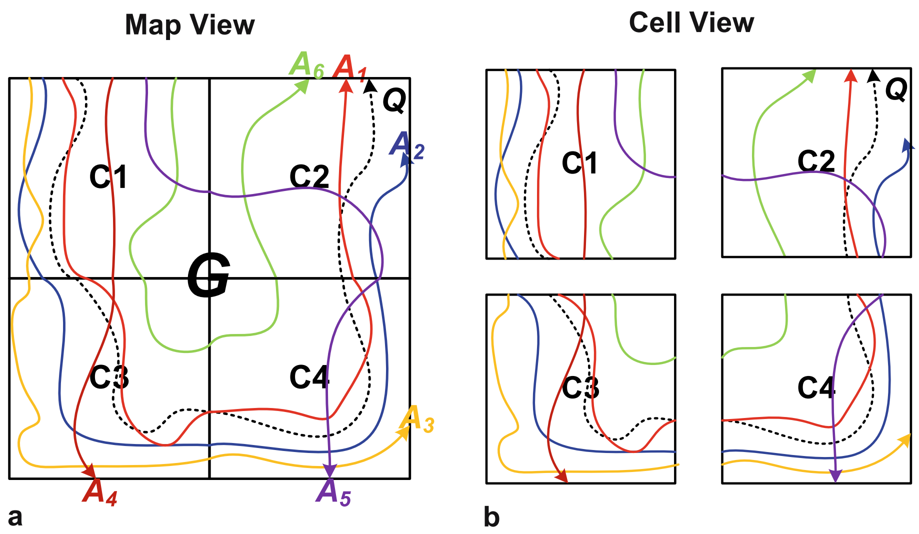
Example. Assume that we have a set of five sensors (s_1, s_2, s_3, s_4 and s_5), and that each sensor maintains locally the scores of five objects o_1, o_2, o_3, o_4 and o_5 (e. g. the sensor measurements at five consecutive time moments). In Fig. 2, we print the local scores of the five objects (ordered by object id). The task is to find the one object which maximizes the average of local scores across all sensors (i. e. o_3).

Distributed Top-k Trajectory Retrieval

Let G denote a 2-dimensional matrix of *points* in the *xy*-plane that represents the coordinate space of some geographic area. Without loss of generality, we assume that the points in G are logically organized into $x \cdot y$ *cells*. Each cell contains an *access point (AP)* that is assumed to be in communication range from every point in its cell.[1]
Although the coordinate space is assumed to be partitioned in square cells, other geometric shapes such as variable size rectangles or Voronoi polygons are similarly applicable. This partitioning of the coordinate space simply denotes that in our setting, G is covered by a set of *APs*. Now let $A_1, A_2, \ldots, A_m$ denote a set of m objects moving in G. At each discrete time instance, object A_i ($\forall i \leq m$) generates a spatio-temporal record $r = \{A_i, t_i, x_i, y_i\}$, where t_i denotes the timestamp on which the record was generated, and (x_i, y_i) the coordinates of A_i at t_i. The record r is then stored locally at the closest *AP* for l discrete time moments after which it is discarded. Therefore at any given point every access point *AP* maintains locally the records of the last l time moments.
The *Distributed Top-K Trajectory Retrieval* problem is defined as follows: *Given a trajectory Q, retrieve the K trajectories which are the most similar to Q.*

Example. Assume that some monitoring area G, is segmented into 4 cells, and that each cell is monitored by an access point (see Fig. 3). A trajectory can be conceptually thought of as a continuous sequence

[1] The terms *access point* and *cell* are used interchangeably.



Top-k Retrieval Techniques in Distributed Sensor Systems, Figure 3 Six spatially fragmented trajectories $A_1, \ldots, A_6$ across four cells $C1$, $C2$, $C3$, $C4$

$A_i = ((a_{x:1,y:1}), \ldots, (a_{x:l,y:l}))$ $(i \leq m)$, while physically it is spatially fragmented across several cells. Similarly, the spatio-temporal query is also represented as: $Q = ((q_{x:1,y:1}), \ldots, (q_{x:l,y:l}))$ but this sequence is not spatially fragmented. The objective is to find the two trajectories that are most similar to the query Q (i. e. A_1 and A_2 for Fig. 3).

Historical Background

There has been a lot of work in the area of Top-K query processing in the database community [2,6,7]. However, most of these solutions have been developed for a centralized database management scenario, where the data is stored on local storage. In this section we will focus our attention on distributed Top-K algorithms, in which the data is accessible over a single-hop, or multi-hop, communication network. Top-K query processing over a multi-hop network is of particular importance in this work, firstly because the multi-hop network is the fundamental assumption underlying the operation of many distributed environments and secondly, because it has not been addressed in the past.

The *Fagin Algorithm (FA)* [2], is one of the first Top-K query processing algorithms over middleware systems which interact with a number of autonomous data sources. In *FA*, the query node *QN* performs a two phase retrieval which consists of a sorted access and random access phase. Initially, *QN* accesses the n lists in parallel until it locates k objects which belong to all lists. In the random phase, *QN* requests from each node to send the score for any object whose score could not be computed exactly. A main problem with FA, is that the initial sorted phase accesses all lists at the same depth. This results in the retrieval of a large number of unnecessary object scores. Additionally, FA cannot be executed efficiently in single-hop or multi-hop environment, because the sorted phase is iterative. As a result, each iteration translates into a large number of communication messages.

The most widely recognized algorithm for Top-K queries in a centralized environment is the *Threshold Algorithm (TA)* [2]. *TA* starts out by performing a parallel sorted access to the n lists row by row. While an object o_i is seen by *QN*, *TA* performs a random access to the other lists to find the exact score for o_i (i. e. $\sum_{j=1}^{n} o_{ij}$). After finding the exact score for each object in the current row[2], it computes a *threshold* value τ as the sum of all scores in the current row. If τ is larger than the kth highest exact score we have seen, the *TA* algorithm performs another iteration in which the threshold τ is refined as the sum of scores across the next row. The algorithm stops after k objects have been found with a score above τ. While the *TA* algorithm accesses less objects than *FA*, it also uses more round trips as it invokes several small random accesses. This would again translate into an arbitrary large number of phases, which is highly undesirable for a distributed environment.

Top-k algorithms have also been studied in other settings where the pruning of the retrieval space is highly desirable. Bruno et al. [6] discuss the problem of answering top-k queries over web accessible databases. The very recent work in [7], examines the problem of approximate top-k queries in distributed environments. It assumes that each node maintains an approximation of the local scores instead of the actual scores. The approximation essentially consists of an equi-width histogram on the local scores

[2]Sorted access is executed on a row-at-a-time basis

along with a bloom filter per histogram bucket which captures object identifiers inserted into the specific bucket. The TPUT [1] algorithm proposed by Cao and Wang, uses three phases in order to resolve top-k queries in distributed settings. The algorithm constructs a uniform bound for data retrieval in all lists, which is able to save communication overhead for query processing. The bound, however, is uniform for all lists, similarly to FA, which is too coarse for data pruning in practice.

On the other hand, spatiotemporal queries have been an intense area of research over the last years [4,11]. This resulted in the development of efficient access methods [4,11] and similarity measures [11] for predictive [10], historical [11] and complex spatio-temporal queries [3]. All these techniques, as well as the frameworks for spatio-temporal queries [9], work in a completely centralized setting. Our techniques on the other hand are decentralized and keep the data *in-situ*, which is more appropriate for environments with expensive communication mediums and for large scale applications that generate huge amounts of spatiotemporal records.

Scientific Fundamentals

We will present efficient algorithms to solve the Distributed Top-K Retrieval problem and the Distributed Top-K Trajectory Retrieval problem in sensor networks.

The TJA Algorithm

We now introduce TJA (Threshold Join Algorithm) [13], which is an efficient top-k query processing algorithm for Sensor Networks. For clarity, we will refer to the collection of local scores at each node v_i, as list(v_i). TJA decreases the number of objects that are required to be transmitted from each list(v_i), by using an additional probing and filtering phase. In addition, the algorithm seeks to optimize the use of the network resources by pushing computation into the network. More specifically, the algorithm consists of three phases: i) the *Lower Bound* phase, in which the querying node finds a lower bound on the lists by probing the nodes in a network ii) the *Hierarchical Joining* phase, in which each node uses the lower bound for eliminating the objects that are below this bound and join the qualifying objects with results coming from children nodes and iii) the *Clean-Up* phase, in which the actual top-k results are identified. The details of the TJA algorithm are described in Algorithm 1.

In the Distributed Top-K Trajectory Retrieval problem, the similarity query Q is initiated by some querying node QN, which disseminates Q to all the cells that intersect the query Q. We call the intersecting cells *candidate cells*. Upon receiving Q, each candidate cell executes locally a lower bounding matching function *(LowerM)* and an upper bounding function *(UpperM)* on all its local subsequences. This yields $2 \cdot m$ local distance computations to Q by each cell (one for each bound). To speed up computations we could utilize spatiotemporal access methods similar to those proposed in [11]. The conceptual array of lower (LB) and upper bounds (UB) for an example scenario of three nodes (C_1, C_2, C_3) is illustrated in Fig. 4a. We will refer to the sum of bounds from all cells as *METADATA* and to the actual subsequence trajectories stored locally by each cell as *DATA*. Obviously, *DATA* is orders of magnitudes more expensive than *METADATA* to be transferred towards *QN*. Therefore we want to intelligently exploit *METADATA* to identify the subset of *DATA* that produces the *K* highest ranked answers. Figure 4b illustrates two typical topologies between cells: star and hierarchy. Our proposed algorithms are equivalently applicable to both of

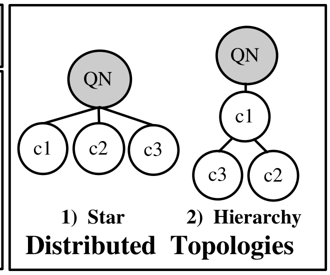
Input: n distributed upper bound lists, parameter K.
Output: The K objects with the highest average upper bounds.

1. QN performs a sorted access to the n lists in parallel (see Figure 4, left side). For each list it retrieves the identifiers of the K (locally) highest ranked objects (that have the highest upper bound). Let $M'[i]$ $(i \leq n)$ denote the answer from each cell. QN creates the union $\tau = \bigcup_{i=1}^{n} M'[i]$.
2. QN performs another sorted access to the n lists in parallel. For each list it retrieves the objects above the threshold τ (i.e. the objects with a score above the minimum score in τ). Let $M''[i]$ $(i \leq n)$ denote the answer from each list. While QN receives the answers, it performs an incremental addition and finally creates the set $R = \bowtie_{i=1}^{n} M''[i]$. An object in R is tagged as *Incomplete* if it was computed using less than n results; or *Complete* otherwise. Therefore $R = \{\{Complete\}\{Incomplete\}\}$.
3. If $K \leq |Complete|$ then stop; else QN performs a random access to the n lists to find the complete result for all *Incomplete* objects. The objects with the K highest results are finally returned as the result.

Top-k Retrieval Techniques in Distributed Sensor Systems, Algorithm 1 TJA (Distributed TOP-K Algorithm)

C1 id,lb,ub	C2 id,lb,ub	C3 id,lb,ub	METADATA id,lb,ub
A2,3,6	A4,4,5	A4,1,3	A4,10,18
A0,4,8	A2,5,6	A0,6,10	A2,13,19
A4,5,10	A0,5,7	A2,5,7	A0,15,25
A7,7,9	A3,5,6	A9,6,7	A3,20,27
A3,8,11	A9,8,10	A3,7,10	A9,22,26
A9,8,9	A7,12,13	A7,11,13	A7,30,35
....	....	....	....

(rows: m; columns: n)



Top-k Retrieval Techniques in Distributed Sensor Systems, Figure 4 *Left METADATA:* Lower and Upper bounds computed for *m* trajectories. *Right* Distributed Topologies

them although we use a star topology here to simplify our description.

In order to find the K trajectories that are most similar to a query trajectory Q, *QN* can fetch all the *DATA* and then perform a centralized similarity computation using the $\text{FullM}(Q, A_i)$ $(\forall i \leq m)$ method, which is one of the LCSS, DTW or other L_p-Norm distance measures. *Centralized* is extremely expensive in terms of data transfer and delay.

Next we present two novel distributed query processing algorithms, *UB-K* and *UBLB-K*, which find the K most similar trajectories to a query trajectory Q. The UB-K algorithm uses an upper bound on the matching between Q and a target trajectory A_i, while UBLB-K uses both a lower and an upper bound on the matching. The description on how these bounds are acquired will be described later.

The UB-K Algorithm

The *UB-K* algorithm [12], is an iterative algorithm for retrieving the K most similar trajectories to a query Q. The algorithm minimizes the number of *DATA* entries transferred towards *QN* by exploiting the upper bounds from the *METADATA* table. Notice that *METADATA* contains the bounds of many objects that will not be in the final top-K result. In order to minimize the cost of uploading the complete *METADATA* table to *QN* we utilize a distributed top-K query processing algorithm, such as *TPUT* [1] or *TJA* [13], that transfers only the most necessary entries from the *METADATA* table towards *QN*. Recall that these algorithms do the following function: Given a set of n distributed scores for each of m objects, they return the K objects with the highest score across all n sites. Note that in our setting, these scores are the local upper bounds. Here we utilize the TJA algorithm, although any other distributed top-k algorithm is applicable as well.

Description. Algorithm 2 presents *UB-K*. In the first step of the algorithm, *QN* retrieves the Λ highest UBs with the assistance of a distributed top-K algorithm. The parameter Λ expresses the user confidence in the *METADATA* bounds. For example when the user is confident that the

Input: Query Q, m Distributed Trajectories, Result Parameter K, Iteration Step λ.
Output: K trajectories with the largest match to Q.

1. Run any distributed top-K algorithm for Q and find the Λ $(\Lambda > K)$ trajectories with the highest UBs.
2. Fetch the $(\Lambda - 1)$ trajectories from the cells and compute their full matching to Q using $FullM(Q, A_i)$.
3. If the Λth UB is smaller or equal to the Kth largest full match then stop; else goto step 1 with $\Lambda{=}\Lambda + \lambda$.

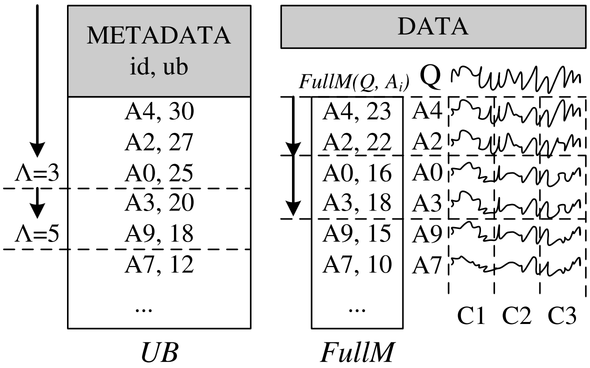
Top-k Retrieval Techniques in Distributed Sensor Systems, Algorithm 2 UB-K Algorithm

METADATA table contains tight bounds, then Λ might be set to a small value.[3] In the second step, *QN* fetches the exact trajectory (i. e. *DATA*) for $\Lambda - 1$ trajectories A_i identified in the first step. It then performs a local computation of $\text{FullM}(Q, A_i)$ and determines their full matching to the query Q. If the Λth highest UB is smaller or equal to the Kth highest full matching value, we terminate with the final result. Otherwise, we perform another iteration by increasing the parameter Λ by λ.

Example. Consider the example scenario of Fig. 5. Assume that the query is to find the top-2 trajectories ($K = 2$) across $C1, C2$ and $C3$. In the first step, *QN* computes the UBs of the highest Λ *METADATA* entries (i. e. A_4, A_2, A_0) using a distributed top-k algorithm. These entries are the ones that have the highest average UB across the three cells. The entries below A_0 in the *METADATA* table are not available to the querying node *QN* at this point. In the second step *QN* will fetch the subsequences of the trajectories identified in the first step. Therefore *QN* has now the complete trajectories for A_4 and A_2 (right side of Fig. 5). *QN* then computes the following full matching: $\text{FullM}(Q, A_4) = 23$, $\text{FullM}(Q, A_2) = 22$ using the Longest Common Subsequence (LCSS). Since the Λth highest UB ($A_0 = 25$) is

[3]Here we initialize Λ as $K + 1$ and set λ as K.

T



Top-k Retrieval Techniques in Distributed Sensor Systems, Figure 5
Example execution of *UB-K*

larger than the Kth highest full match ($A_2 = 22$), the termination condition is not satisfied in the third step. To explain this, consider a trajectory X with a UB of 24 and a full match of 23. Obviously X is not retrieved yet (because it has a smaller UB than 25). However, it is a stronger candidate for the top-2 result than $(A_2, 22)$, as X has a full match of 23 which is larger than 22. Therefore we initiate the second iteration of the UB-K algorithm in which we compute the next λ ($\lambda = 2$) *METADATA* entries and full values FullM $(Q, A_0) = 16$, FullM $(Q, A_3) = 18$. Now the termination has been satisfied because the Λth highest UB $(A_9, 18)$ is smaller than the Kth highest full match $(A_2, 22)$. Finally we return as the top-2 answer the trajectories with the highest full matches (i. e. $\{(A_4, 23), (A_2, 22)\}$).

Theorem 1. *The UB-K algorithm always returns the most similar objects to the query trajectory Q.*

Proof: Let A denote some arbitrary object returned as an answer by the UB-K algorithm ($A \in$ Result), and B some arbitrary object that is not among the returned results ($B \notin$ Result). We want to show that FullM $(Q, B) \leq$ FullM (Q, A) always holds. Assume that FullM $(Q, B) >$ FullM (Q, A). We will show that such an assumption leads to a contradiction. Since $A \in$ Result and $B \notin$ Result it follows from the first step of the algorithm that $ub_B \leq ub_A$. In the second phase of the algorithm we fetch the trajectory A and calculate FullM (Q, A). By using the assumption, we can now draw the following conclusion: FullM $(Q, A) <$ FullM $(Q, B) \leq ub_B \leq ub_A$. When the algorithm terminates in the third step, with A among its answers, we know that ub_X, for some object X, was smaller or equal to the Kth largest full score (i. e. $ub_X \leq \ldots \leq$ FullM (Q, A)). But it is also true that $ub_B \leq ub_X$ (as object B was not chosen in the first step of the algorithm), which yields $ub_B \leq ub_X \leq$ FullM (Q, A) and subsequently FullM $(Q, B) \leq$ FullM (Q, A) (by definition FullM $(Q, B) \leq ub_B$). This is a contradiction as we assumed that FullM $(Q, B) >$ FullM (Q, A).

In addition to the *UB-K* algorithm, we can also utilize both upper bounds and lower bounds to find the top-k results (*UBLB-K* [12]). In contrast with *UB-K*, *UBLB-K* uses the lower bounds instead of the exact scores to determine whether enough candidate results have been retrieved, and it transfers the candidate trajectories in a final bulk step rather than incrementally.

Computing the Bounds

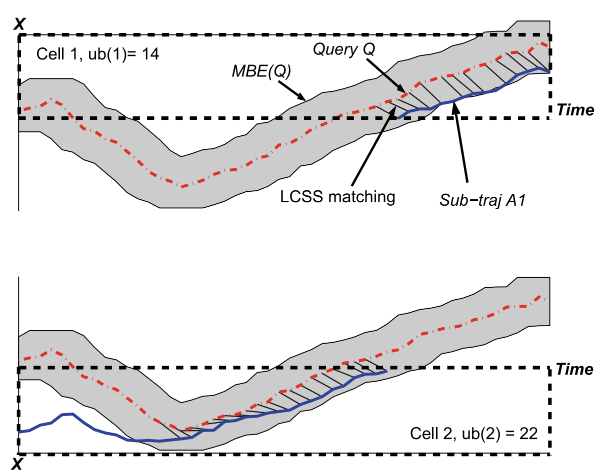
In the following, we describe the techniques for computing upper bounds and lower bounds for various distance measures (i. e. Longest Common SubSequence (*LCSS*) and Dynamic Time Wrapping (*DTW*)). The correctness of these techniques has been proved in [12].

Distributed LCSS Upper Bound *(DUB_LCSS)* Algorithm 3 presents the distributed upper bound algorithm on *LCSS* (*DUB_LCSS*) [12]. The idea is to have each cell c_j

Input: n distributed cells, m trajectories per cell (each denoted as A_{ij} $(i \leq m,\ j \leq n)$), query trajectory Q.
Output: $DUB\_LCSS_{\delta,\epsilon}(MBE_Q, A_i)$, $\forall i \leq m$

1. Each cell c_j $(j \leq n)$ uses $LCSS(MBE_Q, A_{ij})$ to calculate the similarity of each local subsequence trajectory A_{ij} to MBE_Q.
2. The upper bound $DUB\_LCSS_{\delta,\epsilon}(MBE_Q, A_i)$ $(i \leq m)$, is constructed by adding the n local results (i.e. $\sum_{j=1}^{n} LCSS(MBE_Q, A_{ij})$).

Top-k Retrieval Techniques in Distributed Sensor Systems, Algorithm 3 DUB_LCSS Algorithm



Top-k Retrieval Techniques in Distributed Sensor Systems, Figure 6
The local upper bounds of the projections of two cells in the X dimension. The overlap of the sub-trajectory in the cell with the *MBE_Q* is taken as upper bound matching

($\forall j \leq n$) locally match its local subsequences A_{ij} ($\forall i \leq m$) to Q using the upper bounding method $LCSS\,(MBE_Q, A_{ij})$ described in [11]. Note that this is a simple and cheap operation since each trajectory point in the local subsequence A_{ij} is associated with a timestamp. We then simply perform a parallel addition of these individual results which yields an upper bound on the LCSS matching. Figure 6 illustrates the operations of the algorithm.

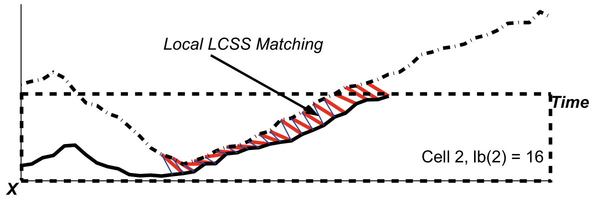
Distributed LCSS Lower Bound *(DLB_LCSS)* Algorithm 4 presents the distributed lower bound algorithm on *LCSS* [12]. The idea is again to perform n distributed computations with a local similarity function, and then perform a parallel addition of these individual results. Our lower bound is computed by having each cell c_j ($j \leq n$), to perform a local computation of $LCSS_{\delta,\varepsilon}(Q'_{ij}, A_{ij})$ ($\forall i \leq m$, $\forall j \leq n$), without extending the warping window δ outside A_{ij}. Figure 7 illustrates the computation of the local lower bound.

Input: n distributed cells, m trajectories per cell (each denoted as A_{ij} ($i \leq m$, $j \leq n$)), query trajectory Q.
Output: $DLB\_LCSS_{\delta,\epsilon}(Q, A_i)$, $\forall i \leq m$

1. For each trajectory A_i, cell c_j ($j \leq n$) finds the time region $T_{ij} = \{ts(p)|p \in A_{ij}\}$ when A_i stays in cell c_j. Filter Q into Q'_{ij} such that Q'_{ij} is in the same time intervals as A_{ij}, $Q'_{ij} = \{p|p \in Q \; and \; ts(p) \in T_{ij}\}$.
2. Each cell c_j ($j \leq n$) performs a local computation of $LCSS_{\delta,\epsilon}(Q'_{ij}, A_{ij})$ ($\forall i \leq m, \forall j \leq n$).
3. The lower bound $DLB\_LCSS_{\delta,\epsilon}(Q, A_i)$ ($\forall i \leq m$), is constructed by adding the n local results (i.e. $\sum_{j=1}^{n} LCSS_{\delta,\epsilon}(Q'_{ij}, A_{ij})$).

Top-k Retrieval Techniques in Distributed Sensor Systems, Algorithm 4 DLB_LCSS Algorithm

Bounding Dynamic Time Wrapping (DTW) Distance Our distributed upper and lower bounding algorithms for LCSS can be extended to bound the DTW distance of two trajectories. Below we sketch our approach. Since DTW is a distance measure, to find the most similar trajectory we have to find the trajectory with the smallest DTW distance. We can use a Minimum Bounding Envelop (*MBE*) to lower bound the DTW distance [11]. In other words, we have that $DTW_{\delta,\varepsilon}\,(Q, A) > DTW_{\delta,\varepsilon}\,(MBE_Q, A)$. The DTW distance can be lower bounded by adding together all the local lower bound distances in each cell (analogously to the computation of the upper bound for LCSS similarity). On the other hand, to upper bound the DTW distance, we perform a similar distributed computation as for computing



Top-k Retrieval Techniques in Distributed Sensor Systems, Figure 7 The local lower bound computation lb (Q, A) in the projection of cell 2 in the X dimension. The *thick red line* represents full LCSS matching, while the *thin blue line* represents the lower bound matching within the cell

DLB_LCSS. The difference is that each cell applies DTW as a local computation function instead of LCSS (i. e. Algorithm 4 can be used to upper bound the DTW distance if we replace $LCSS_{\delta,\varepsilon}\,(Q'_{ij}, A_{ij})$ with $DTW_{\delta,\varepsilon}\,(Q'_{ij}, A_{ij})$). This implies that we can come up with algorithms analogously to UB-K and UBLB-K to find the trajectories that are the most similar (i. e. they minimize the DTW distance to the query Q).

Future Directions

The future work includes developing techniques that will allow people to answer efficiently continuous, rather than sporadic, distributed Top-K queries. Another possible future work would be to investigate the effectiveness of the framework under a number of different real life scenarios.

Recommended Reading

1. Cao, P., Wang, Z.: Efficient top-k query calculation in distributed networks. In: PODC 04: Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing, pp. 206–215, St. John's, Newfoundland, Canada, 25–28 July 2004
2. Fagin, R.: Combining fuzzy information from multiple systems (extended abstract). In: PODS 96: Proceedings of the fifteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, pp. 216–226, Montreal, Canada, 3–5 June 1996
3. Hadjieleftheriou, M., Kollios, G., Bakalov, P., Tsotras, V.J.: Complex spatio-temporal pattern queries. In: VLDB 05: Proceedings of the 31st international conference on Very large data bases, pp. 877–888, Trondheim, Norway, 30 Aug–2 Sept 2005
4. Kollios, G., Gunopulos, D., Tsotras, V.J.: On indexing mobile objects. In: PODS 99: Proceedings of the eighteenth ACM SIGMODSIGACT- SIGART symposium on Principles of database systems, pp. 261–272, Philadelphia, Pennsylvania, 31 May–2 June 1999
5. Madden, S., Franklin, M.J., Hellerstein, J.M., Hong, W.: Tag: a tiny aggregation service for ad-hoc sensor networks. ACM SIGOPS Operating Systems Review **36**(SI), 131–146 (2002)
6. Marian, A., Bruno, N., Gravano, L.: Evaluating top-k queries over web-accessible databases. ACM Transactions on Database Systems **29**(2), 319–362 (2004)

T

7. Michel, S., Triantafillou, P., Weikum, G.: Klee: a framework for distributed top-k query algorithms. In VLDB 05: Proceedings of the 31st international conference on Very large data bases, pp. 637–648, Trondheim, Norway, 30 Aug–2 Sept 2005
8. Nieto, M.: Public video surveillance: Is it an effective crime prevention tool? Technical report, California Research Bureau Report, June 1997
9. Saltenis, S., Jensen, C.S., Leutenegger, S.T., Lopez, M.A.: Indexing the positions of continuously moving objects. In: SIGMOD 00: Proceedings of the 2000 ACM SIGMOD international conference on Management of data, pp. 331–342, Dallas, Texas, 16–18 May 2000
10. Tao, Y., Sun, J., Papadias, D.: Analysis of predictive spatio-temporal queries. ACM Trans Database Syst **28**(4), 295–336 (2003)
11. Vlachos, M., Hadjieleftheriou, M., Gunopulos, D., Keogh, E.: Indexing multi-dimensional time-series with support for multiple distance measures. In KDD 03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 216–225, Washington, D.C., USA, 24–27 Aug 2003
12. Zeinalipour-Yazti, D., Lin, S., Gunopulos, D.: Distributed spatio-temporal similarity search. In: CIKM 06: Proceedings of the 15th ACM international conference on Information and knowledge management, pp. 14–23, Arlington, VA, USA, 6–11 Nov 2006
13. Zeinalipour-Yazti, D., Vagena, Z., Gunopulos, D., Kalogeraki, V., Tsotras, V., Vlachos, M., Koudas, N., Srivastava, D.: The threshold join algorithm for top-k queries in distributed sensor networks. In: DMSN 05: Proceedings of the 2nd international workshop on Data management for sensor networks, pp. 61–66, Trondheim, Norway, 29 Aug 2005

Top-N Queries

► Skyline Queries

Topological Operators

► Dimensionally Extended Nine-Intersection Model (DE-9IM)

Toponym

► Retrieval Algorithms, Spatial

TPR-Trees

► Indexing Schemes for Multi-dimensional Moving Objects
► Indexing, BDual Tree
► Movement Patterns in Spatio-temporal Data
► R*-tree

Tracking

► Indoor Localization

Trajectories, Discovering Similar

GEORGE KOLLIOS[1], MICHAIL VLACHOS[2], DIMITRIS GUNOPULOS[3]

[1] Department of Computer Science, Boston University, Boston, MA, USA
[2] IBM T.J. Watson Research Center, Yorktown Heights, NY, USA
[3] Department of Computer Science and Engineering, University of California, Riverside, CA, USA

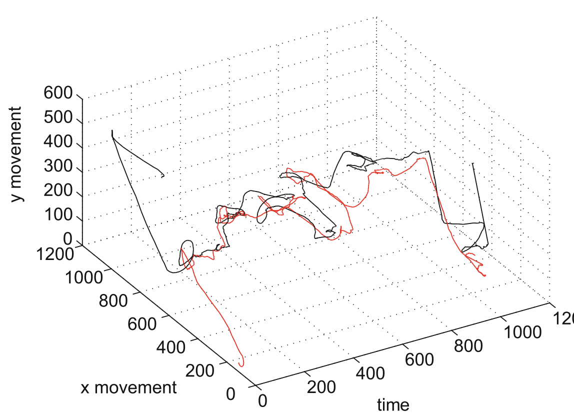
Synonyms

Multi-dimensional time series similarity; Mining spatio-temporal datasets

Definition

The trajectory of a moving object is typically modeled as a sequence of consecutive locations in a multi-dimensional (generally two or three dimensional) Euclidean space. Such data types arise in many applications where the location of a given object is measured repeatedly over time. Typical trajectory data are obtained during a tracking procedure with the aid of various sensors. Here also lies the main obstacle of such data; they may contain a significant amount of *outliers* or in other words incorrect data measurements (unlike for example, stock data which contain no errors whatsoever). An example of two trajectories is shown in Fig. 1.

Many data mining tasks, such as clustering and classification, necessitate a *distance function* that is used to estimate the *similarity* or *dis-similarity* between any two objects



Trajectories, Discovering Similar, Figure 1 Examples of 2D trajectories in the database. Furthermore, in order to provide efficient solutions to many data mining tasks, a method that can retrieve quickly the data objects that are more similar to a given query object (or a set of objects) is required. Therefore, to perform data mining on trajectories of moving objects, the following problem must be addressed: given a database $\mathcal{D}$ of trajectories and a query $\mathcal{Q}$ (not already in the database), the system has to find the trajectory $\mathcal{T}$ that is closest to $\mathcal{Q}$. In order to solve this problem, two important sub-problems must be addressed: i) define a realistic and appropriate distance function and ii) design an indexing scheme for answering the nearest neighbor query efficiently.

Historical Background

Trajectories are modeled as multi-dimensional time series. Most of the related work on time-series data analysis has concentrated on the use of some metric L_p norm. The L_p norm distance between two n-dimensional vectors $\bar{x}$ and $\bar{y}$ is defined as $L_p(\bar{x}, \bar{y}) = (\sum_{i=1}^{n}(|x_i - y_i|)^p)^{1/p}$. For $p = 2$ it is the well known Euclidean distance and for $p = 1$ the Manhattan distance. The advantage of this simple metric is that it allows efficient indexing with a dimensionality reduction technique [1,7,10]. On the other hand, the model cannot deal well with outliers and is very sensitive to small distortions in the time axis [15]. There are a number of interesting extensions to the above model to support various transformations such as scaling [14], shifting [8], normalization [8] and moving average [14].

Other techniques to define time series similarity are based on extracting certain features (landmarks [12] or signatures [6]) from each time-series and then using these features to define the similarity. Another approach is to represent a time series using the direction of the sequence at regular time intervals [13].

Although the vast majority of database/data mining research on time series data mining has focused on Euclidean distance, virtually all real world systems that use time series matching as a subroutine, utilize a similarity measure which allows warping. In retrospect, this is not very surprising, since most real world processes, particularly biological processes, can evolve at varying rates. For example, in bioinformatics, it is well understood that functionally related genes will express themselves in similar ways, but possibly at different rates. Therefore, the Dynamic Time Warping (DTW) distance has been used for many datasets of this type. The method to compute DTW between two sequences is based on Dynamic Programming [2] and is more expensive than computing L_p norms. Approaches to mitigate the large computational cost of the DTW have appeared in [9,16] where lower bounding functions are used in order to speed up the execution of DTW. Furthermore, an approach to combine the benefits of warping distances and L_p norms has been proposed in [4].

The flexibility provided by DTW is very important, however its efficiency deteriorates for noisy data, since by matching all the points, it also matches the outliers distorting the true distance between the sequences. An alternative approach is the use of *Longest Common Subsequence* (*LCSS*), which is a variation of the edit distance [11]. The basic idea is to match two sequences by allowing them to stretch, without rearranging the order of the elements but allowing some elements to be *unmatched*. Using the *LCSS* of two sequences, one can define the distance using the length of this subsequence [3,5].

Scientific Fundamentals

First, some definitions are provided and then the similarity functions based on the appropriate models are presented. It is assumed that objects are points that move on the *(x,y)*-plane and time is discrete.

Let A and B be two trajectories of moving objects with size n and m respectively, where $A = ((a_{x,1}, a_{y,1}), \ldots, (a_{x,n}, a_{y,n}))$ and $B = ((b_{x,1}, b_{y,1}), \ldots, (b_{x,m}, b_{y,m}))$. For a trajectory A, let *Head(A)* be the sequence $Head(A) = ((a_{x,1}, a_{y,1}), \ldots, (a_{x,n-1}, a_{y,n-1}))$.

Given an integer δ and a real number $0 < \epsilon < 1$, the $LCSS_{\delta,\epsilon}(A, B)$ is defined as follows:

$$LCSS_{\delta,\epsilon}(A,B) = \begin{cases} 0, & \text{if A or B is empty} \\ 1 + LCSS_{\delta,\epsilon}(Head(A), Head(B)), & \text{if } |a_{x,n} - b_{x,m}| < \epsilon \text{ and } |a_{y,n} - b_{y,m}| < \epsilon \text{ and } |n-m| < \delta \\ \max(LCSS_{\delta,\epsilon}(Head(A), B),\ LCSS_{\delta,\epsilon}(A, Head(B))), & \text{otherwise}. \end{cases}$$

The constant δ controls how far in time can go in order to match a given point from one trajectory to a point in another trajectory. The constant ε is the matching threshold (see Fig. 2).

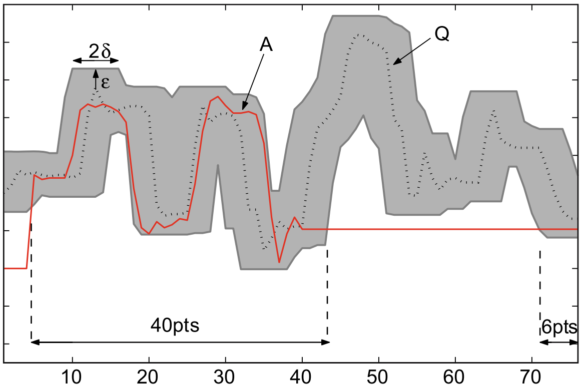
The first similarity function is based on the *LCSS* and the idea is to allow time stretching. Then, objects that are close in space at different time instants can be matched if the time instants are also not very far.

Therefore, the similarity function *S1* between two trajectories A and B, given δ and ε, is defined as follows:

$$S1(\delta, \epsilon, A, B) = \frac{LCSS_{\delta,\epsilon}(A, B)}{\min(n, m)}.$$

The division by the length of the sequence in $S1$ serves the purpose of comparing the LCSS value between sequences of different lengths.

T



Trajectories, Discovering Similar, Figure 2 The notion of the *LCSS* matching within a region of δ & ε for a trajectory. The points of the 2 trajectories within the *gray region* can be matched by the *LCSS* function

The $S1$ function is used to define another similarity measure that is more suitable for trajectories. Consider the set of all translations. A translation simply shifts a trajectory in space by a different constant in each dimension. Let $\mathcal{F}$ be the family of translations. Then a function $f_{c,d}$ belongs to $\mathcal{F}$ if $f_{c,d}(A) = ((a_{x,1}+c, a_{y,1}+d), \ldots, (a_{x,n}+c, a_{y,n}+d))$. Using this family of translation, the following distance function is defined.

Given δ, ε and the family $\mathcal{F}$ of translations, the similarity function $S2$ between two trajectories A and B, is defined as follows:

$$S2(\delta, \epsilon, A, B) = \max_{f_{c,d} \in \mathcal{F}} S1(\delta, \epsilon, A, f_{c,d}(B)) .$$

The similarity functions $S1$ and $S2$ range from 0 to 1. Therefore, the distance function between two trajectories can be estimated as follows:

Given δ, ε and two trajectories A and B, then:

$$D1(\delta, \epsilon, A, B) = 1 - S1(\delta, \epsilon, A, B) \quad \text{and}$$
$$D2(\delta, \epsilon, A, B) = 1 - S2(\delta, \epsilon, A, B) .$$

Note that $D1$ and $D2$ are *symmetric*. $LCSS_{\delta,\epsilon}(A, B)$ is equal to $LCSS_{\delta,\epsilon}(B, A)$ and the transformation that is used in $D2$ is a translation which preserves the symmetric property.

By allowing translations, similarities between movements that are parallel in space can be detected. In addition, the *LCSS* model allows stretching and displacement in time, so it can detect similarities in movements that happen with different speeds, or at different times.

Given the definitions above, efficient methods to compute the distance functions are presented next.

Computing the Similarity Function $S1$

To compute the similarity functions $S1$, $S2$ an *LCSS* computation is needed. The *LCSS* can be computed by a dynamic programming algorithm in $O(n^2)$ time. However, if matchings are allowed only when the difference in the indices is at most δ, a faster algorithm is possible. The following result has been shown in [2,5]: Given two trajectories A and B, with $|A| = n$ and $|B| = m$, the $LCSS_{\delta,\epsilon}(A, B)$ can be found in $O(\delta(n + m))$ time.

If δ is small, the dynamic programming algorithm is very efficient. However, for some applications, δ may need to be large. For that case, the above computation can be improved using random sampling.

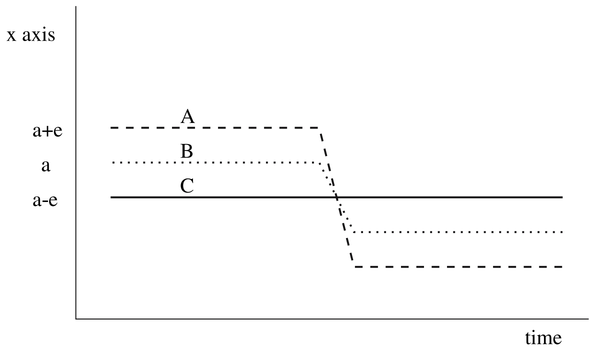
By taking a sufficiently small amount of random samples from the original data, it can be shown that with high probability the random sample preserves the properties (shape, structure, average value, etc) of the original population. The random sampling method will give an approximate result but with a probabilistic guarantee on the error. In particular, it can be shown that, given two trajectories A and B with length n, two constants δ and ε, and a random sample of A, $|RA| = s$, an approximation of the $LCSS(\delta, \epsilon, A, B)$ can be computed such that the approximation error is less than β with probability at least $1 - \rho$, in $O(ns)$ time, where $s = f(\rho, \beta)$. To give a practical perspective of the random sampling approach, to be within 0.1 of the true similarity of two trajectories A and B, with a probability of 90% and the similarity between them is around 0.8, the A should be sampled at 250 locations. Notice that this number is independent of the length of both A and B. To be able to capture accurately the similarity between less similar trajectories (e. g. with 0.4 similarity) then more sample points must be used (e. g. 500 points).

Computing the Similarity Function $S2$

Consider now the more complex similarity function $S2$. Here, given two sequences A,B, and constants δ, ϵ, the translation $f_{c,d}$ that maximizes the length of the longest common subsequence of $A, f_{c,d}(B)$ ($LCSS_{\delta,\epsilon}(A, f_{c,d}(B))$ over all possible translations must be found.

Let the length of trajectories A and B be n and m respectively. Let also assume that the translation f_{c_1,d_1} is the translation that, when applied to B, gives a longest common subsequence: $LCSS_{\delta,\epsilon}(A, f_{c_1,d_1}(B)) = a$, and it is also the translation that maximizes the length of the longest common subsequence: $LCSS_{\delta,\epsilon}(A, f_{c_1,d_1}(B)) = \max_{c,d \in \mathcal{R}} LCSS_{\delta,\epsilon}(A, f_{c,d}(B))$.

The key observation is that, although there is an infinite number of translations that can be applied on B, each translation $f_{c,d}$ results in a longest common subsequence between A and $f_{c,d}(B)$, and there is a finite set of possible longest common subsequences. Therefore, it is possible to enumerate the set of translations, such that this set provably includes a translation that maximizes the length of the



Trajectories, Discovering Similar, Figure 3 An example where the triangle inequality does not hold for the $D2$ distance

longest common subsequence of A and $f_{c,d}(B)$. Based on this idea, it has been shown in [15] that: Given two trajectories A and B, with $|A| = n$ and $|B| = m$, the $S2(\delta, \epsilon, A, B)$ can be computed in $O((n+m)^3\delta^3)$ time.
Furthermore, a more efficient algorithm has been proposed that achieves a running time of $O((m+n)\delta^3/\beta^2)$, given a constant $0 < \beta < 1$. However, this algorithm is approximate and the approximation $AS2_{\delta,\beta}(A, B)$ is related with the actual distance with the formula: $S2(\delta, \epsilon, A, B) - AS2_{\delta,\beta}(A, B) < \beta$.

Indexing for LCCS Based Similarity

Even though the approximation algorithm for the $D2$ distance significantly reduces the computational cost over the exact algorithm, it can still be costly when one is interested in similarity search on massive trajectory databases. Thus, a hierarchical clustering algorithm using the distance $D2$ is provided that can be used to answer efficiently similarity queries.
The major obstacle in providing an indexing scheme for the distance function $D2$ is that $D2$ is not a metric, since it does not obey the triangle inequality. This makes the use of traditional indexing techniques difficult. Indeed, it is easy to construct examples with trajectories A, B and C, where $D2(\delta, \epsilon, A, C) > D2(\delta, \epsilon, A, B) + D2(\delta, \epsilon, B, C)$. Such an example is shown in Fig. 3, where $D2(\delta, \epsilon, A, B) = D2(\delta, \epsilon, B, C) = 0$ (since the similarity is 1), and $D2(\delta, \epsilon, A, C) = 1$ (because the similarity within ε in space is zero).
However, a weaker version of the triangle inequality can be proven, which can help prunning parts of the database and improve the search performance. First, the following function is defined:

$$LCSS_{\delta,\epsilon,\mathcal{F}}(A, B) = \max_{f_{c,d}\in\mathcal{F}} LCSS_{\delta,\epsilon}(A, f_{c,d}(B)) .$$

Clearly, $D2(\delta, \epsilon, A, B) = 1 - \frac{LCSS_{\delta,\epsilon,\mathcal{F}}(A,B)}{min(|A|,|B|)}$ (as before, $\mathcal{F}$ is the set of translations). Now, the following can be shown: Given trajectories A,B,C:

$$\begin{aligned} LCSS_{\delta,2\epsilon,\mathcal{F}}(A, C) \geq\ & LCSS_{\delta,\epsilon,\mathcal{F}}(A, B) \\ & + LCSS_{\delta,\epsilon,\mathcal{F}}(B, C) - |B| \end{aligned}$$

where $|B|$ is the length of sequence B.
To create the indexing structure, the set of trajectories is partitioned into groups according to their length, so that the longest trajectory in each group is at most a times the shortest (typically $a = 2$ is used.) Then, a hierarchical clustering algorithm is applied on each set, and the tree that the algorithm produces is used as follows:
For every node C of the tree, the medoid (M_C) of the cluster represented by this node is stored. The medoid is the trajectory that has the minimum distance (or maximum LCSS) from every other trajectory in the cluster: $\max_{v_i\in C} \min_{v_j\in C} LCSS_{\delta,\epsilon,\mathcal{F}}(v_i, v_j, e)$. However, keeping only the medoid is not enough. Note that, a method is needed to efficiently prune part of the tree during the search procedure. Namely, given the tree and a query sequence Q, the algorithm should decide whether whether to follow the subtree that is rooted at C or not. However, from the previous lemma it is known that for any sequence B in C:

$$\begin{aligned} LCSS_{\delta,\epsilon,\mathcal{F}}(B, Q) < |B| + & LCSS_{\delta,2\epsilon,\mathcal{F}}(M_C, Q) \\ & - LCSS_{\delta,\epsilon,\mathcal{F}}(M_C, B) \end{aligned}$$

or in terms of distance:

$$\begin{aligned} & D2(\delta, \epsilon, B, Q) \\ & = 1 - \frac{LCSS_{\delta,\epsilon,\mathcal{F}}(B, Q)}{\min(|B|, |Q|)} \\ & > 1 - \frac{|B|}{\min(|B|, |Q|)} \\ & \quad - \frac{LCSS_{\delta,2\epsilon,\mathcal{F}}(M_C, Q)}{\min(|B|, |Q|)} + \frac{LCSS_{\delta,\epsilon,\mathcal{F}}(M_C, B)}{\min(|B|, |Q|)} . \end{aligned}$$

In order to provide an upper bound on the similarity (or a lower bound on the distance) the expression $|B| - LCSS_{\delta,\epsilon,\mathcal{F}}(A, B)$ must be maximized. Therefore, for every node of the tree along with the medoid the trajectory r_c that maximizes this expression is stored. Using this trajectory a lower bound on the distance between the query and any trajectory on the subtree can be estimated.
Next, the search function that uses the index structure discussed above is presented. It is assumed that the tree contains trajectories with minimum length *minl* and maximum length *maxl*. For simplicity, only the algorithm for the 1-Nearest Neighbor query is presented.
The search procedure takes as input a node N in the tree, the query Q and the distance to the closest trajectory found

T

```
Procedure Search(N, Q, mindist)
for each children C of N do {
  if C is a leaf { /* singleton cluster */
    if (D2(C,Q) < mindist) {
      mindist = D2(C, Q);
      NNtraj = C;
      return;
    }
  } else { /* non-leaf, cluster with multiple trajectories */
    if (|C| < minl) {
      K =|C|
    } else if (|C| > maxl) {
      K = |r_C|
    } else {
      K = minl
    }
    L = 1 − (|r_C|+LCSS_{δ,2ε,F}(M_C,r_C)−LCSS_{δ,2ε,F}(M_C,Q)) / K
    if (mindist > L)
      Search(C, Q, mindist)
    else
      return;
}
```

Trajectories, Discovering Similar, Figure 4 Search procedure for nearest neighbor queries

so far (Fig. 4). For each of the children C, it is checked if it is a trajectory or a cluster. In case that it is a trajectory, its distance to Q is compared with the current nearest trajectory. If it is a cluster, first the length of the query is checked and then the appropriate value for $\min(|B|, |Q|)$ is chosen. Thus, a lower bound L is computed on the distance of the query with any trajectory in the cluster and the result is compared with the distance of the current nearest neighbor *mindist*. This cluster is examined only if L is smaller than *mindist*. In the scheme above, the approximate algorithm to compute the $LCSS_{\delta,\epsilon,\mathcal{F}}$ is used. Consequently, the value of $(LCSS_{\delta,\epsilon,\mathcal{F}}(M_C, B))/(\min(|B|, |Q|))$ that is computed can be up to β times higher than the exact value. Therefore, since the approximate algorithm of section 3.2 is used, the $(\beta * \min(|M_C|, |B|))/(\min(|B|, |Q|))$ should be subtracted from the bound for $D2(\delta, \epsilon, B, Q)$ to get the correct results.

Key Applications

Sciences

Trajectory data with the characteristics discussed above (multi-dimensional and noisy) appear in many scientific data. In environmental, earth science and biological data analysis, scientists may be interested in identifying similar patterns (e. g. weather patterns), cluster related objects or subjects based on their trajectories and retrieve subjects with similar movements (e. g., in animal migration studies). In medical applications similar problems may occur, for example, when multiple attribute response curves in drug therapy are analyzed.

Transportation and Monitoring Applications

In many monitoring applications, detecting movements of objects or subjects that exhibit similarity in space and time can be useful. These movements may have been reconstructed from a set of sensors, including cameras and movement sensors and therefore are inherently noisy. Another set of applications arise from cell phone and mobile communication applications where mobile users are tracked over time and patterns and clusters of these users can be used for improving the quality of the network (i. e., by allocating appropriate bandwidth over time and space).

Future Directions

So far, it is assumed that objects are points that move in a multi-dimensional space, ignoring their shape in space. However, there are many applications where the extent of each object is also important. Therefore, a future direction, is to design similarity models for moving objects with extents, when both the locations and the extents of the objects change over time.
Another direction is to design a more general indexing scheme for distance functions that are similar to *LCCS* and can work for multiple distance functions and datasets.

Cross References

► Co-location Pattern
► Patterns, Complex

Recommended Reading

1. Agrawal, R., Faloutsos, C., Swami, A.: Efficient similarity search in sequence databases. In Proc. of FODO, pp. 69–84 (1993)
2. Berndt, D., Clifford, J.: Using dynamic time warping to find patterns in time series. In AAAI Workshop on Knowledge Discovery in Databases, pp. 229–248 (1994)
3. Bollobás, B., Das, G., Gunopulos, D., Mannila, H.: Time-series similarity problems and well-separated geometric sets. In Proc. of SCG (1997)
4. Chen, L., Ng, R.T.: On the marriage of lp-norms and edit distance. In Proc. of VLDB, pp. 792–803 (2004)
5. Das, G., Gunopulos, D., Mannila, H.: Finding Similar Time Series. In Proc. of PKDD, pp. 88–100 (1997)
6. Faloutsos, C., Jagadish, H.V., Mendelzon, A., Milo, T.: Signature technique for similarity-based queries. In Proc. of SEQUENCES (1997)
7. Faloutsos, C., Ranganathan, M., Manolopoulos, I.: Fast subsequence matching in time series databases. In Proc. of ACM SIGMOD, May (1994)
8. Goldin, D., Kanellakis, P.: On similarity queries for time-series data. In Proc. of CP '95, September (1995)
9. Keogh, E.: Exact indexing of dynamic time warping. In Proc. of VLDB (2002)

10. Keogh, E., Chakrabarti, K., Mehrotra, S., Pazzani, M.: Locally adaptive dimensionality reduction for indexing large time series databases. In Proc. of ACM SIGMOD, pp. 151–162 (2001)
11. Levenshtein, V.: Binary codes capable of correcting deletions, insertions, and reversals. In Soviet Physics – Doklady 10, 10, pp. 707–710 (1966)
12. Perng, S., Wang, H., Zhang, S., Parker, D.S.: Landmarks: a new model for similarity-based pattern querying in time series databases. In Proc. of IEEE ICDE, pp. 33–42 (2000)
13. Qu, Y., Wang, C., Wang, X.S.: Supporting fast search in time series for movement patterns in multiple scales. In Proc. of ACM CIKM, pp. 251–258 (1998)
14. Rafiei, D., Mendelzon, A.: Querying time series data based on similarity. IEEE Transactions on Knowledge and Data Engineering, Vol. 12, No 5., pp. 675–693 (2000)
15. Vlachos, M., Kollios, G., Gunopulos, D.: Discovering similar multidimensional trajectories. In Proc. of IEEE ICDE, pp. 673–684 (2002)
16. Zhu, Y., Shasha, D.: Query by humming: a time series database approach. In Proc. of ACM SIGMOD (2003)

Trajectory

RALF HARTMUT GÜTING
Fakultät für Mathematik und Informatik,
Fernuniversität Hagen, Hagen, Germany

Synonyms

Moving points

Definition

Representation of a time dependent position observed over some period of time. Usually represented as a polyline in a 3D (2D + time) space for an object moving in the 2D plane.

Main Text

Trajectories describe complete histories of movement; they are stored in *moving objects databases*, sometimes called *trajectory databases* in the literature. When operations are included, a trajectory corresponds to a value of a *moving point* data type. Queries of databases containing trajectories can be formulated using *moving object languages*. When uncertainty about an object's precise position is taken into account, an uncertain trajectory [3] results which can be viewed as a kind of cylindrical volume in the 2D + time space. There exists much work on indexing trajectories [2]. Querying for trajectories similar to a given one is also an important research area [1].

Cross References

► Moving Object Languages
► Spatio-temporal Data Types

Recommended Reading

1. Chen, L., Özsu, M.T., Oria, V.: Robust and fast similarity search for moving object trajectories. Proc. ACM SIGMOD Conf., Baltimore, Maryland, USA, pp. 491–502 (2005)
2. Mokbel, M.F., Ghanem, T.M., Aref, W.G.: Spatio-temporal access methods. Bull. of the TC on Data Engineering **26**(2), 40–49 (2003)
3. Trajcevski, G., Wolfson, O., Hinrichs K., Chamberlain, S.: Managing uncertainty in moving objects databases. ACM Transactions on Database Systems (TODS) **29**(3), 463–507 (2004)

Trajectory Patterns

► Movement Patterns in Spatio-temporal Data

Transportation Network Model

► Road Network Data Model

Travel Time Computation

► Dynamic Travel Time Maps

Traveling Salesman Problem (TSP)

RHETT WILFAHRT[1], SANGHO KIM[2]
[1] Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA
[2] ESRI, Redlands, CA, USA

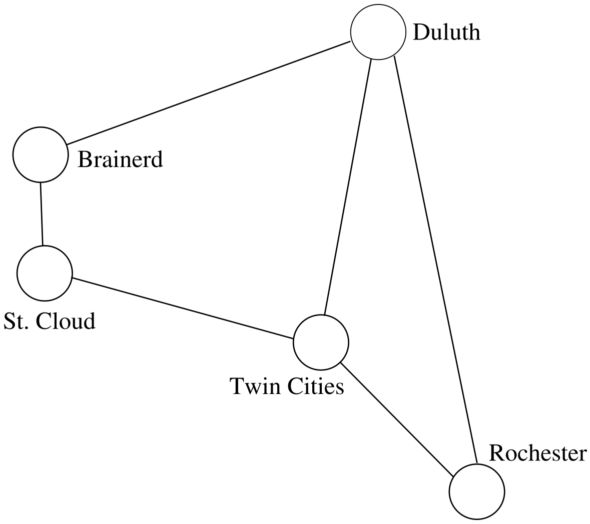
Synonyms

Traveling salesperson problem; TSP; Hamiltonian cycle with the least weight

Definition

The traveling salesperson problem is a well studied and famous problem in the area of computer science. In brief, consider a salesperson who wants to travel around the country from city to city to sell his wares. A simple example is shown in Fig. 1.

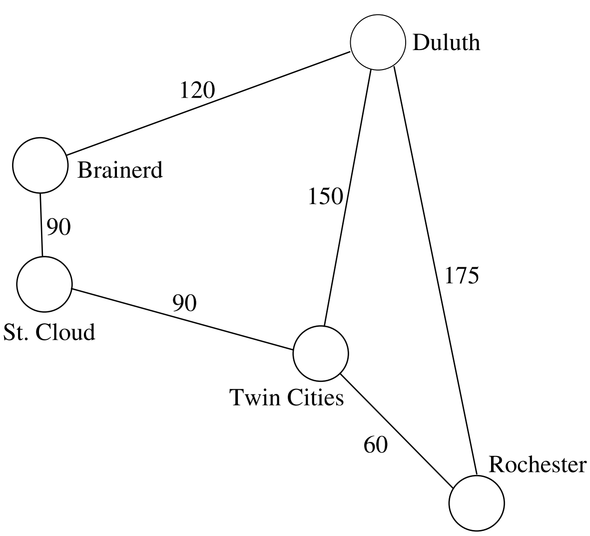
The traveling salesperson does not want to visit any city twice and at the end of his trip he wants to return to the same city he started in. The question is what route can the salesperson take to exhaustively visit all the cities without going through the same city twice. It is also in the salesperson's best interest to spend the least amount of time traveling, therefore, he would like to cover the least possible total distance. In order to facilitate this need, distance or travel time should be incorporated in the map as edge cost. More formally, the problem can be considered as a connected graph where the nodes are the cities and the edges are the roads between them. Each edge has a weight associated with it that is analogous to a distance. The goal of



Traveling Salesman Problem (TSP), Figure 1 An example of a city map for the traveling salesman problem

the problem is to visit all of the nodes without visiting any twice and do this while traveling the edges but incurring the minimum cost.

To see what a solution may look like to this problem, let us consider the diagram in Fig. 2 If a salesman starts in Brainerd, he can travel to St. Cloud, then to the Twin Cities, then to Rochester, then to Duluth and finally back to Brainerd. The total cost for this trip would be $90+90+60+175+120=525$. The simplicity of this example makes it easy to see that the solution is an optimal one. For a graph as sparsely connected as this one,



Traveling Salesman Problem (TSP), Figure 2 An example of a city map with cost on edges

the solutions will often be fairly easy to see; the problem grows in complexity as the graphs become more densely connected.

It should be noted that there is an area of research in mathematics that is analogous to the traveling salesperson problem called Hamiltonian cycles. A Hamiltonian cycle is a cycle that visits each node of a graph exactly once and returns to the original spot on the graph. Finding the optimal Hamiltonian cycle is equivalent to solving the traveling salesperson problem.

Historical Background

The Irish mathematician William Hamilton and the British mathematician Thomas Kirkman started studies on graph theory problems which were closely related to the TSP problem in the 1800's [9]. They formulated the problem as follows. Given a complete weighted graph, find a Hamiltonian cycle with the least weight. Other mathematicians, Karl Menger, Hassler Whitney and Merrill Flood, helped in further development of the problem in the 1900's and established the problem as one of the most famous within the combinatorial optimization field.

Scientific Fundamentals

The traveling salesperson problem is of great theoretical importance. It falls into a class of problems known as NP-hard problems. It can be shown that all problems within this class of problems are fundamentally related, such that solving one of them with a reasonable cost would allow us to solve them all with a reasonable cost. A naïve approach to solving this problem through a brute force methodology is of O(n!) complexity, which is completely inadequate for any real problem sizes. With the brute force method, all possible permutations are enumerated and the cheapest one is selected. There are two approaches to this problem. The first is to develop good heuristics to approximate solutions to the problem in reasonable complexity. The second, the holy grail, is to solve the problem in a reasonable complexity. Current methods focused on a heuristic solution are often based on mathematical approaches. A simple example of a heuristic approach is a naïve greedy algorithm that will always take the path with the least cost. If the heuristic is used in our example of drilling holes in the circuit board, it is likely that the location closest to the last drill location will be chosen regardless of the overall distribution of the drill holes. This heuristic, also known as the nearest neighbor heuristic, is an example of a construction heuristic. It is important to note that heuristics are not designed to give us an optimal solution, only to quickly get very close. Heuristics often have a very aggressive running time with non-exponential growth patterns. This makes them very attractive when compared to the exponential growth of any exhaustive searching technique for a problem of any non-trivial size.

Another common heuristic approach to solving the traveling salesperson problem heuristically is to solve a simpler form of the problem. The portions of the problem can be constrained such that it becomes easier. A classic example is the circuit board problem. If a constraint is added to specify the given map as a fully connected graph with weights based on Euclidean distance, it is much easier to get some traction on the problem. The hope is then that the solution can be leveraged to a simpler problem in solving the more general problem.

The second approach is to solve the problem for optimal solution with a reasonable complexity. While a naïve algorithm to come up with an optimal solution is trivial it becomes very hard or impossible to discover optimal solutions in anything less than exponential time. Algorithms using branch-and-bound methods are known to solve the TSP problem which consists of around 50 cities. Linear programming approaches can solve the problem with more than 200 cities. There have been many trials to solve the TSP with an order of magnitude since the year 2000 due to the recent advancement in computer technology. The most recent report on the successful solution was a problem size of more than tens of millions of cities.

The high profile nature of the traveling salesperson problem and the need for heuristic solutions has resulted in threads of research. This leads to the need for benchmark datasets for a variety of heuristic algorithms to compare the datasets against. In addition, the benchmark must be large enough to pose a computational problem to the heuristics such that their run time growth can be analyzed, but small enough so that the problem can be optimally solved to determine how close a heuristic solution was to the optimal. This provides two measuring sticks for heuristic solutions: running time and proximity to the optimal solution.

A large number of datasets are available, with the most obvious source being the GIS domain. Any number of cities can be taken in the world to form arbitrary large traveling salesperson problems. The unfortunate truth is that it is difficult to establish any lasting benchmark for this problem because the "trivial" boundary moves very quickly as new more powerful computers are developed.

Key Applications

This innocuous problem turns out to have many applications. The main application is a logistical one related directly to the movement of something in space to accomplish some sort of a task. A classic example is that of school buses within a school district. A graph of bus stop locations can be established that are interconnected by the roads with weights signifying distance or time between each node. Then, it is possible to find the least cost solution to picking up all of the school children. Many commercial service providers such as FedEx take advantage of the solutions of TSPin order to reduce logistical costs such as fuel consumption and travel time. Another example is that of scheduling drill holes in a circuit board. The locations of all the holes are identified, represented by the nodes, allowing the graph to be fully connected with weights according to the Euclidean distance.

Besides the logistics and circuit design, many other types of applications can be found such as the clustering of data arrays, the sequencing of jobs on a single machine, material handling in a warehouse, the analysis of crystal structure, the overhauling of gas turbine engines, cutting stock problems, and the assignment of routes for planes of a specified fleet. In addition, the TSP is involved in applications as variations and sub-problems. One example of such variations is the resource constrained traveling salesman problem which has applications in scheduling with an aggregate deadline. The prize collecting traveling salesman problem and the orienteering problem are special cases of the resource constrained TSP.

Future Directions

Many approaches to the traveling salesperson problem and their descriptions can be found in the literature. For a decent overview, refer to [4]. The solution to this problem (and the related class of problems) is still open and in fact now has a prize purse associated with it, although the prize purse is technically for solving any of the NP-hard class of problems. This prize is hosted by the Clay Mathematics Institute and this is one of their "millennium problems" [3]. As previously noted, select large datasets have been solved for this problem, but in general, it takes tremendous amounts of computing power to solve such large datasets.

In the electrical engineering industry, optimized circuit design using TSP is of great importance. As the circuit is more compactly condensed with a tremendous number of nodes and edges, an efficient solution of the TSP is actively pursued. In the delivery industry, it is of main concern to reduce logistical costs by optimizing the routes of delivery vehicles. Many GIS software companies like ESRI are trying to provide an efficient solution of the TSP in combination with GIS dataset.

Cross References

► Fastest-Path Computation
► Graph Theory, Konigsberg Problem

T

► Nearest Neighbors Problem
► Routing Vehicles, Algorithms

Recommended Reading

1. Traveling Salesman Problem website supported by School of Industrial and Systems Engineering at Georgia Tech. http://www.tsp.gatech.edu/
2. Wikipedia, Traveling Salesman Problem. http://en.wikipedia.org/
3. Millennium Problems, The Clay Mathematics Institute of Cambridge, Massachusetts. http://www.claymath.org/millennium/
4. Appletgate, D., Bixby, R., Chvátal, V., Cook, W.: On the solution of traveling salesman problems. Documenta Mathematica Journal der Deutschen Mathematiker-Vereinigung International Congress of Mathematicians, (1998)
5. Fekete, S., Meijer, H., Rohe, A., Tietze, W.: Solving a Hard problem to approximate an Easy one: heuristics for maximum matchings and maximum traveling salesman problems. J. Exp. Algorithm. (JEA)**7**, (2002)
6. Barvinok, A., Fekete, S., Johnson, D., Tamir, A., Woeginger, G., Woodroofe, R.: The geometric maximum traveling salesman problem. J. ACM (JACM) **50**:(5)3, (2003)
7. Krolak, P., Felts, W., Marble, G.: A man-machine approach toward solving the traveling salesman problem. Communications of the ACM **14**:(5), (1971)
8. Papadimitriou, C., Steiglitz, K.: Some complexity results for the Traveling Salesman Problem. Proceedings of the eighth annual ACM symposium on Theory of computing, May (1976)
9. Biggs, N.L., Lloyd, E.K., Wilson, R.J.: Graph Theory 1736–1936. Clarendon Press, Oxford (1976)

Traveling Salesperson Problem

► Traveling Salesman Problem (TSP)

Trend-Surface Analysis

► Geographic Dynamics, Visualization And Modeling

Trend Surfaces

► Constraint Databases and Data Interpolation

Triage, Query

► Retrieval Algorithms, Spatial

Triangulated Irregular Network

► Photogrammetric Products

Triangulation

► Indoor Localization

Trichotomous Theory of Landmarks

► Wayfinding, Landmarks

Trilateration

► Indoor Localization

Trip Planning

► Trip Planning Queries in Road Network Databases

Trip Planning Queries in Road Network Databases

FEIFEI LI[1], MARIOS HADJIELEFTHERIOU[2], GEORGE KOLLIOS[3], DIHAN CHENG[3], SHANG-HUA TENG[3]

[1] Department of Computer Science, Florida State University, Tallahassee, FL, USA
[2] AT&T Labs Inc, Florham Park, NJ, USA
[3] Computer Science Department, Boston University, Boston, MA, USA

Synonyms

Optimal sequenced route query; Multi-type nearest neighbor query; Spatial road networks; Trip planning; Generalized minimum spanning tree; Nearest neighbor search

Definition

Consider a database that stores a spatial road network and a set of points of interest that are located on the edges of the network and belong to one or more categories from a fixed set of categories C. The Trip Planning Query (TPQ) is defined as follows: the user specifies two points on the edges of the network, a starting point S and a destination point E, and a subset of categories $\mathcal{R}$, ($\mathcal{R} \subseteq C$), and the goal is to find the *best* trip (route) that starts at S, passes through exactly one point from each category in $\mathcal{R}$ and ends at E. An example of a TPQ is the following: A user plans to travel from Boston to Providence and wants to stop at a supermarket, a bank, and a post office. Given this query, a database that stores the locations of objects from the categories above (as well as other categories) should compute efficiently a feasible trip that minimizes the total traveling distance. Another possibility is to provide a trip that minimizes the total traveling time.

To formalize it further, consider the TPQ problem on a *metric graphs*. Given a connected graph $G(\mathcal{V}, \mathcal{E})$ with n vertices $\mathcal{V} = \{v_1, \ldots, v_n\}$ and s edges $\mathcal{E} = \{e_1, \ldots, e_s\}$, the cost of traversing a path $v_i, \ldots, v_j$ is denoted by $c(v_i, \ldots, v_j) \geq 0$. G is a metric graph if it satisfies the following conditions: 1) $c(v_i, v_j) = 0$ iff $v_i = v_j$; 2) $c(v_i, v_j) = c(v_j, v_i)$; and 3) The triangle inequality $c(v_i, v_k) + c(v_k, v_j) \geq c(v_i, v_j)$. Given a set of m categories $C = \{C_1, \ldots, C_m\}$ (where $m < qn$) and a mapping function $\pi: v_i \longrightarrow C_j$ that maps each vertex $v_i \in \mathcal{V}$ to a category $C_j \in C$, the TPQ problem can be defined as follows:

Given a set $\mathcal{R} \subseteq C$ ($\mathcal{R} = \{R_1, R_2, \ldots, R_k\}$), a starting vertex S and an ending vertex E, identify the vertex traversal $\mathcal{T} = \{S, v_{t_1}, \ldots, v_{t_k}, E\}$ (also called a trip) from S to E that visits at least one vertex from each category in $\mathcal{R}$ (i. e., $\cup_{i=1}^{k} \pi(v_{t_i}) = \mathcal{R}$) and has the minimum possible cost $c(\mathcal{T})$ (i. e., for any other feasible trip $\mathcal{T}'$ satisfying the condition above, $c(\mathcal{T}) \leq c(\mathcal{T}')$).

Historical Background

TPQ was proposed in [7]. It can be considered as a generalization of the Traveling Salesman problem (TSP) [1] which is *NP*-hard. The reduction of TSP to TPQ is straightforward. By assuming that every point belongs to its own distinct category, any instance of TSP can be reduced to an instance of TPQ. A simple polynomial time 2-approximation algorithm for TSP on a metric graph can be obtained using the Minimum Spanning Tree (MST) [4]. The best constant approximation ratio for metric TSP is the $\frac{3}{2}$-approximation that can be derived by the Christofides algorithm [3]. Also, a polynomial time approximation scheme (PTAS) for Euclidean TSP has been proposed by Arora [1]. For any fixed $\varepsilon > 0$ and any n nodes in $\mathbb{R}^2$ the randomized version of the scheme can achieve a $(1 + \varepsilon)$-approximation in $O(n \log^{O(\frac{1}{\varepsilon})} n)$ running time. There are many approximation algorithms for variations of the TSP problem, e. g., TSP with neighborhoods [5], nevertheless these problems are not closely related to TPQ queries. A very good reference for a number of approximation algorithms on different versions of TSP is [1]. Finally, there are many practical heuristics for TSP [13], e. g., genetic and greedy algorithms, that work well for some practical instances of the problem, but no approximation bounds are known about them. The optimal sequenced route selection problem proposed in [11] is the same problem but with the additional constraint that the user specifies the order of the visited categories. Therefore, the problem is not NP-hard anymore and can be solved in polynomial time. Another similar problem appeared in [8], where the start and end points of the TPQ are the same point. The algorithm presented in [8] is based on R-trees and is focused on objects located in a Euclidean space.

TPQ is also closely related to the Generalized Minimum Spanning Tree (GMST) problem. The GMST is a generalized version of the MST problem where the vertices in a graph G belong to m different categories. A tree T is a GMST of G if T contains at least one vertex from each category and T has the minimum possible cost (total weight or total length). Even though the MST problem is in P, it is known that the GMST is in NP. There are a few methods from the operational research and economics community that propose heuristics for solving this problem [9] without providing a detailed analysis on the approximation bounds. The GMST problem is a special instance of an even harder problem, the Group Steiner Tree (GST) problem [6]. Since the GMST problem is a special instance of the GST problem, such bounds apply to GMST as well.

Scientific Fundamentals

Because TPQ is difficult to solve optimally (NP-hard problem), approximation algorithms for answering the query are examined. For ease of exposition, a solution for the case that objects can move anywhere in space (not on a specific network) is presented first and then this solutin is extended to the road network case.

In the rest, the total number of vertices is denoted by n, the total number of categories by m, and the maximum cardinality of any category by ρ. For simplicity, it will be assumed that $\mathcal{R} = C$, thus $k = m$. Generalizations for $\mathcal{R} \subset C$ are straightforward. Also, $\mathcal{T}_a^P$ denotes an approximation trip for problem P, while $\mathcal{T}_o^P$ denotes the optimal trip. When P is clear from context the superscript is dropped.

T

Nearest Neighbor Algorithm

The most intuitive algorithm for solving TPQ is to form a trip by iteratively visiting the nearest neighbor of the last vertex added to the trip from all vertices in the categories that have not been visited yet, starting from S. Formally, given a partial trip $\mathcal{T}_k$ with $k < m$, $\mathcal{T}_{k+1}$ is obtained by inserting the vertex $v_{t_{k+1}}$ which is the nearest neighbor of v_{t_k} from the set of vertices in $\mathcal{R}$ belonging to categories that have not been covered yet. In the end, the final trip is produced by connecting v_{t_m} to E. This algorithm is called $\mathcal{A}_{NN}$, which is shown in Algorithm 1.

Algorithm 1 $\mathcal{A}_{NN}(G^c, \mathcal{R}, S, E)$
1: $v = S, I = \{1, \ldots, m\}, \mathcal{T}_a = \{S\}$
2: **for** $k = 1$ to m **do**

$v =$ the nearest $NN(v, R_i)$ for all $i \in I$
$\mathcal{T}_a \leftarrow \{v\}$
$I \leftarrow I - \{i\}$
end for
$\mathcal{T}_a \leftarrow \{E\}$

It is possible to bound the approximation ratio of algorithm $\mathcal{A}_{NN}$. Theorem 1 gives its bound. It should be pointed out that such bound is obtained based on the worst case analysis. In practice, much better bound will be obtained for a given problem instance.

Theorem 1. *$\mathcal{A}_{NN}$ gives a $(2^{m+1} - 1)$-approximation (with respect to the optimal solution). In addition, this approximation bound is tight.*

Minimum Distance Algorithm

The main advantage of $\mathcal{A}_{NN}$ is its simplicity and efficiency. However, the main disadvantage of $\mathcal{A}_{NN}$ is the problem of "searching without directions". Consider the example shown in Fig. 1a. $\mathcal{A}_{NN}$ will find the trip $T1 = \{S \rightarrow A1 \rightarrow B1 \rightarrow C1 \rightarrow E\}$ instead of the optimal trip $T2 = \{S \rightarrow C2 \rightarrow A2 \rightarrow B2 \rightarrow E\}$. In $\mathcal{A}_{NN}$, the search in every step greedily expands the point that is closest to the last point in the partial trip without considering the end destination, i. e., without considering the direction. The more intuitive approach is to limit the search within a vicinity area defined by S and E as the ellipse shown in Fig. 1b.

Therefore, a novel greedy algorithm is introduced that is based on the intuition discussed above, called $\mathcal{A}_{MD}$, that achieves a much better approximation bound, in comparison with the previous algorithm. The algorithm chooses a set of vertices $\{v_1, \ldots, v_m\}$, one vertex per category in $\mathcal{R}$, such that the sum of costs $c(S, v_i) + c(v_i, E)$ per v_i is the minimum cost among all vertices belonging to the respective category R_i (i. e., this is the vertex from category R_i with the minimum traveling distance from S to E). After the set of vertices has been discovered, the algorithm creates a trip from S to E by traversing these vertices in nearest neighbor order, i. e., by visiting the nearest neighbor of the last vertex added to the trip, starting with S. The algorithm is shown in Algorithm 2.

Algorithm 2 $\mathcal{A}_{MD}(G^c, \mathcal{R}, S, E)$

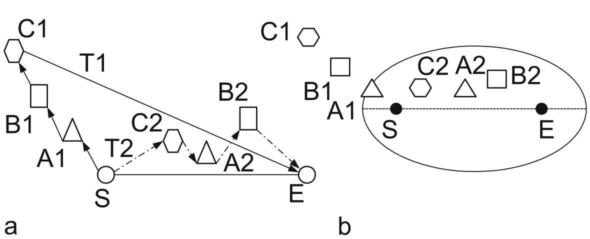
$U = \varnothing$
for $i = 1$ to m **do**
  $U \leftarrow \pi(v) = R_i$: $c(S, v) + c(v, E)$ is minimized
$v = S$, $\mathcal{T}_a \leftarrow \{S\}$
while $U \neq \varnothing$ **do**
  $v = NN(v, U)$
  $\mathcal{T}_a \leftarrow \{v\}$
  Remove v from U
end while
$\mathcal{T}_a \leftarrow \{E\}$

Theorem 2. *If m is odd then $\mathcal{A}_{MD}$ gives an m-approximate solution. In addition this approximation bound is tight. If m is even then $\mathcal{A}_{MD}$ gives an $m+1$-approximate solution. In addition this approximation bound is tight.*

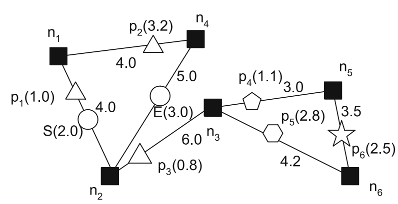
Similarly, the bound for $\mathcal{A}_{MD}$ is based on worst case analysis. In practice, much better bound is expected for any particular problem instance.

TP Queries in Road Networks

Here, TPQ algorithms for road networks datasets are presented. Given a graph $\mathcal{N}$ representing a road network and a separate set $\mathcal{P}$ representing points of interest (gas stations, hotels, restaurants, etc.) located at fixed coordinates on the edges of the graph, appropriate index structures are needed in order to answer efficiently trip planning queries for visiting points of interest in $\mathcal{P}$ using the underlying network $\mathcal{N}$. Figure 2 shows an example road network, along with various points of interest belonging to four different categories. The road network is represented and stored using techniques from [10,12,14]. In summary, the adjacency list of $\mathcal{N}$ and set $\mathcal{P}$ are stored as two separate flat files indexed by B^+-trees. For that purpose, the location of any point $p \in \mathcal{P}$ is represented as an offset from the road network node with the smallest identifier that is incident on the edge containing p. For example, point p_4 is 1.1 units away from node n_3.



Trip Planning Queries in Road Network Databases, Figure 1 Intuition of vicinity area



Trip Planning Queries in Road Network Databases, Figure 2 A simple road network

Implementation of $\mathcal{A}_{NN}$

Nearest neighbor queries on road networks have been studied in [10], where a simple extension of the well known Dijkstra algorithm [4] for the single-source shortest-path problem on weighted graphs is utilized to locate the nearest point of interest to a given query point. The algorithm of [10] can be used to incrementally locate the nearest neighbor of the last stop added to the trip, that belongs to a category that has not been visited yet. The algorithm starts from point S and when at least one stop from each category has been added to the trip, the shortest path from the last discovered stop to E is computed.

Implementation of $\mathcal{A}_{MD}$

Here, the idea is to first locate the m points from categories in $\mathcal{R}$ that minimize the network distance $c(S, p_i, E)$ using the underlying graph $\mathcal{N}$, and then create a trip that traverses all p_i in a nearest neighbor order, from S to E. It is easy to show with a counter example that simply finding a point p that first minimizes cost $c(S, p)$ and then traverses the shortest path from p to E, does not necessarily minimize cost $c(S, p, E)$. Thus, Dijkstra's algorithm cannot be directly applied to solve this problem. Alternatively, an algorithm (shown in Algorithm 3) is proposed for identifying such points of interest.

The algorithm locates a point of interest $p : \pi(p) \in R_i$ (given R_i) such that the distance $c(S, p, E)$ is minimized. The search begins from S and incrementally expands all possible paths from S to E through all points p. Whenever such a path is computed and all other partial trips have cost smaller than the tentative best cost, the search stops. The key idea of the algorithm is to separate partial trips into two categories: one that contains only paths that have not discovered a point of interest yet, and one that contains paths that have. Paths in the first category compete to find the shortest possible route from S to any p. Paths in the second category compete to find the shortest path from their respective p to E. The overall best path is the one that minimizes the sum of both costs.

Algorithm 3 Minimum Distance Query FOR ROAD NETWORKS

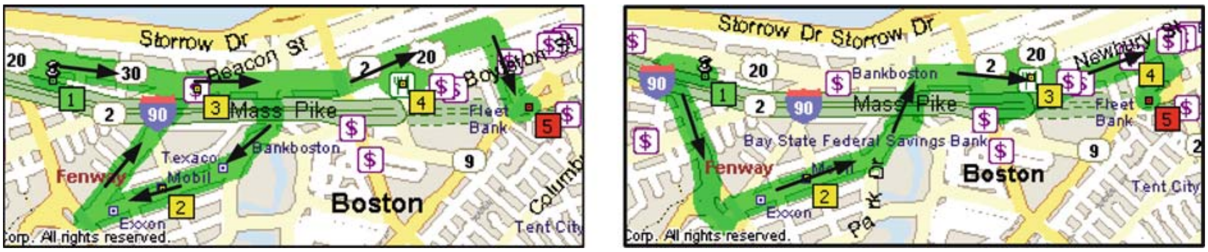
Graph $\mathcal{N}$, Points of interest $\mathcal{P}$, Points S, E, Category R_i
For each $n_\text{i} \in \mathcal{N}$: $n_\text{i}.c_\text{p} = n_\text{i}.c_{\neg\text{p}} = \infty$
PriorityQueue $PQ = \{S\}$, $B = \infty$, $\mathcal{T}_B = \varnothing$
while PQ not empty **do**
    $\mathcal{T} = PQ.top$
    if $\mathcal{T}.c \geq B$ **then** return $\mathcal{T}_B$
    for each node n adjacent to $\mathcal{T}.last$ **do**
        $\mathcal{T}' = \mathcal{T}$ ▷ (create a copy)
        if $\mathcal{T}'$ does not contain a p **then**
            if $\exists p$: $p \in \mathcal{P}$, $\pi(p) = R_\text{i}$ on edge $(\mathcal{T}'.last, n)$ **then**
                $\mathcal{T}'.c += c(\mathcal{T}'.last, p)$
                $\mathcal{T}' \leftarrow p$, $PQ \leftarrow \mathcal{T}'$
            else
                $\mathcal{T}'.c += c(\mathcal{T}'.last, n)$, $\mathcal{T}' \leftarrow n$
                if $n.c_{\neg\text{p}} > \mathcal{T}'.c$ **then**
                $n.c_{\neg\text{p}} = \mathcal{T}'.c$, $PQ \leftarrow \mathcal{T}'$
        else
            if edge $(\mathcal{T}', n)$ contains E **then**
                $\mathcal{T}'.c += c(\mathcal{T}'.last, E)$, $\mathcal{T}' \leftarrow E$
                Update B and $\mathcal{T}_B$ accordingly
            else
                $\mathcal{T}'.c += c(\mathcal{T}'.last, n)$, $\mathcal{T}' \leftarrow n$
                if $n.c_\text{p} > \mathcal{T}'.c$
                $n.c_\text{p} = \mathcal{T}'.c$, $PQ \leftarrow \mathcal{T}'$
        endif
    endfor
endwhile

The algorithm proceeds greedily by expanding at every step the trip with the smallest current cost. Furthermore, in order to be able to prune trips that are not promising, based on already discovered trips, the algorithm maintains two partial best costs per node $n \in \mathcal{N}$. Cost $n.c_p$ ($n.c_{\neg p}$) represents the partial cost of the best trip that passes through this node and that has (has not) discovered an interesting point yet. After all k points (one from each category $R_i \in \mathcal{R}$) have been discovered by iteratively calling this algorithm, an approximate trip for TPQ can be produced.

Next, an example is used to illustrate how the algorithm works. Consider the case shown in Fig. 2. Using the simple Dijkstra based approach, i. e., finding a point p that first minimizes cost $c(S, p)$ and then traverses the shortest path from p to E, will return the trip $S \rightarrow p_1 \rightarrow n_2 \rightarrow E$ with distance of 7.0. However, a better answer is $S \rightarrow n_2 \rightarrow p_3 \rightarrow n_2 \rightarrow E$, which is achieved by the algorithm, with distance of 6.6. In Table 1 the priority queue that contains the search paths along with the update to the node partial best costs is listed in a step by step fashion. The pruning steps of the algorithm make it very efficient. A lot of unnecessary search has been pruned out during the expansions. For example, in step 3 the partial path Sn_2p_1 is pruned out as $c(Sn_2p_1n_1) > n_1.c_p$ and $c(sn_2p_1n_2) > n_2.c_p$. This algorithm can also be used to answer top k queries. Simply maintaining a priority queue for $\mathcal{T}_B$ and update B corresponding to the kth complete path cost. For example if $k = 3$, then in step 4 $\mathcal{T}_B$ will be updated to $Sn_2p_3n_2E(6.6)$, $Sp_1n_2E(7)$ and in step 5 path $Sp_1n_1n_4E(8)$ will be added and the search will stop as by

Trip Planning Queries in Road Network Databases, Table 1 Applying Algorithm 3 to example in Fig. 2

step	priority queue	updates
1	$Sp_1(1)$, $Sn_2(2)$	$n_2.c_{\neg p} = 2$
2	$Sp_1n_1(2)$, $Sn_2p_3(2.8)$, $Sp_1n_2(4)$, $Sn_2p_1(5)$, $Sn_2n_4(7)$	$n_1.c_p = 2$, $n_2.c_p = 4$, $n_4.c_{\neg p} = 7$
3	$Sn_2p_3n_2(3.6)$, $Sp_1n_1n_4(6)$, $Sn_2p_3n_3(8)$, $Sn_2n_4p_2(10.2)$	$n_2.c_p = 3.6$, $n_3.c_p = 8$, $n_4.c_p = 6$,
		$B = 7$, $\mathcal{T}_B = Sp_1n_2E$
4	$Sp_1n_1n_4(6)$, $Sn_2p_3n_3(8)$, $Sn_2n_4p_2(10.2)$	$B = 6.6$, $\mathcal{T}_B = Sn_2p_3n_2E$
5	$Sn_2p_3n_3(8)$, $Sn_2n_4p_3(10.2)$	$B = 6.6$, $\mathcal{T}_B = Sn_2p_3n_2E$
6	Algorithm stops and returns $\mathcal{T}_B$	$B = 6.6$, $\mathcal{T}_B = Sn_2p_3n_2E$



Trip Planning Queries in Road Network Databases, Figure 3 A route from Boston University (1) to Boston downtown (5) that passes by a gas station (2), an ATM (3), and a Greek restaurant (4) that have been explicitly specified by the user in that order. Existing applications do not support route optimization, nor do they give suggestions of more suitable routes, like the one presented *to the right*

now the top partial path has cost equal to the third best complete path.

Key Applications

Efficient TPQ evaluation could become an important new feature of advanced navigation systems and can prove useful for other geographic information systems applications as well. For instance, state of the art mapping services like MapQuest, Google Maps, and Microsoft Streets & Trips, currently support queries that specify a starting point and only one destination, or a number of user specified destinations. The functionality and usefulness of such systems can be greatly improved by supporting more advanced query types, like TPQ. An example from Streets & Trips is shown in Fig. 3, where the user has explicitly chosen a route that includes an ATM, a gas station and a Greek restaurant. Clearly, the system could not only optimize this route by re-arranging the order in which these stops should be made, but it could also suggest alternatives, based on other options available (e. g., from a large number of ATMs that are shown on the map), that the user might not be aware of.

Future Directions

One future direction is to extend the TPQ with additional user specific constraints. For example, one can specify a query, where for each category (or a subset of them) a deadline is set. Visiting a location of a specific category is useful, only if it is done before the specified deadline [2]. For example, someone may want to visit a restaurant between 12 noon and 12:30 PM.

From a theoretical point of view, a good direction is to find a tight lower bound for the main memory version of the problem and solutions that match the approximation ratio indicated by this lower bound. Also, it is interesting to investigate if there is a PTAS for this problem.

Cross References

► Nearest Neighbor Query
► Traveling Salesman Problem (TSP)

Recommended Reading

1. Arora, S.: Polynomial-Time Approximation Schemes for Euclidean TSP and other Geometric Problems. J. ACM **45**(5), 753–782 (1998)
2. Bansal, N., Blum, A., Chawla, S., Meyerson, A.: Approximation Algorithms for Deadline-TSP and Vehicle Routing with Time-Windows. In: Proc. of FOCS (2004)
3. Christofides, N.: Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Computer Science Department, CMU (1976)
4. Cormen, T., Leiserson, C., Rivest, R., Stein, C.: Introduction to Algorithms. The MIT Press (1997)
5. Dumitrescu, A., Mitchell, J.S.B.: Approximation algorithms for tsp with neighborhoods in the plane. In: SODA '01: Proceedings

of the twelfth annual ACM-SIAM symposium on Discrete algorithms, pp. 38–46 (2001)
6. Garg, N., Konjevod, G., Ravi, R.: A Polylogarithmic Approximation Algorithm for the Group Steiner Tree Problem. In: Proc. of SODA (1998)
7. Li, F., Cheng, D., Hadjieleftheriou, M., Kollios, G., Teng., S.-H.: On trip planning queries in spatial databases. In: SSTD: Proc. of the International Symposium on Advances in Spatial and Temporal Databases, pp. 273–290 (2005)
8. Ma, X., Shekhar, S., Xiong, H., Zhang, P.: Exploiting a page-level upper bound for multi-type nearest neighbor queries. In: ACM GIS, pp. 179–186 (2006)
9. Myung, Y., Lee, C., Tcha, D.: On the Generalized Minimum Spanning Tree Problem. Networks (1995)
10. Papadias, D., Zhang, J., Mamoulis, N., Taom, Y.: Query Processing in Spatial Network Databases. In: Proc. of VLDB (2003)
11. Sharifzadeh, M., Kolahdouzan, M., Shahabi, C.: The Optimal Sequenced Route Query. VLDB Journal (2007)
12. Shekhar, S., Liu, D.: CCAM: A connectivity/clustered access method for networks and network computations. IEEE TKDE **19**(1), 102–119 (1997)
13. TSP Home Web Site. http://www.tsp.gatech.edu/
14. Yiu, M., Mamoulis, N.: Clustering Objects on a Spatial Network. In: Proc. of SIGMOD (2004)

TSP

► Traveling Salesman Problem (TSP)

Typology of Landmarks

► Wayfinding, Landmarks

T

UML: Unified Modeling Language

JUGURTA LISBOA FILHO, JOSÉ LUÍS BRAGA
Department of Computer Science/Information Systems, Federal University of Vicosa (UFV), Vicosa, Brazil

Synonyms

Software design model

Definition

Unified Modeling Language (UML) is a language for system specification that was created by several modeling specialists to consolidate and standardize a great number of languages and development methods of object-oriented software appearing in the late 1980s and early 1990s. UML comprises a set of diagrams used to specify, model, construct and document software systems. UML diagrams can be used in both the development of simple systems and in large and complex systems, being also used to model other non software systems.

Main Text

In the middle of the 1990s, three authors of object-oriented software development methods, James Rumbaugh [Object Modeling Technique (OMT)], Grady Booch [Object-Oriented Design (OOD)] and Ivar Jacobson [Object-Oriented Software Engineering (OOSE), also proposed the Use Cases], start a project for unifying their methodologies, working together at Rational Software, creating the UML [1].
In 1997, the UML was adopted as the standard language for software artifact development by the Object Management Group (OMG), which became responsible for the standardization and specification of UML future versions. UML is currently in its second main version and is largely accepted both by the industry and research institutes.
The UML 2.0 [3] consists of 13 types of diagrams, which can be divided into three categories: Structure Diagrams; Behavior Diagrams; and Interaction Diagrams. The Structure Diagrams are used to model system static components and includes the Class Diagram, Object Diagram, Component Diagram, Composite Structure Diagram, Package Diagram, and Deployment Diagram. The Behavior Diagrams are used to model system dynamic aspects and include the Use Case Diagram, Activity Diagram, and State Machine Diagram. Finally, the Interaction Diagrams allow the system interface specification and include the Sequence Diagram, Communication Diagram, Timing Diagram, and Interaction Overview Diagram.
UML is not a method for software construction; on the contrary, UML is independent of any method and can be used with different methods. The methods provide the necessary steps to obtain and to analyze the requirements of an application, integrating them into a system project. The methods for system design are usually supported by computer-aided software engineering (CASE) tools. There are many CASE tools enabling the designer to draw up the several UML diagrams. OMG defined the XML Meta-Data Interchange format (XMI), an extensible markup language (XML) standard for transmission of descriptive metadata of projects, which enables the transfer of UML specifications among different CASE tools [2].
Widely accepted in the field of computer science, the current trend is that UML will become consolidated as a standard for software project specification in general, including the development of geographical information systems (GIS) applications. To attend problems of design in specific domains, the UML can be extended by using its own extension mechanisms.

Cross References

► Unified Modeling Language Extension Mechanisms

Recommended Reading

1. Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language Reference Manual. Addison-Wesley, Reading, MA (1999)
2. Object Management Group.: OMG Unified Modeling Language Specification, Version 1.5. Object Management Group (2003)
3. Object Management Group.: Unified Modeling Language: Superstructure, Version 2.0,.Object Management Group (2005)

UMN MapServer

► University of Minnesota (UMN) Map Server

Uncertain Boundaries

► Objects with Broad Boundaries

Uncertain Environmental Variables in GIS

GERARD B.M. HEUVELINK[1], JAMES D. BROWN[2]
[1] Environmental Sciences Group, Wageningen University, Wageningen, The Netherlands
[2] National Weather Service, National Oceanic and Atmospheric Administration, Silver Spring, MD, USA

Synonyms

Spatial data quality; Attribute and positional error in GIS; Spatial accuracy assessment; Accuracy; Error; Probability theory; Object-oriented; Taylor series; Monte carlo simulation

Definition

Environmental variables are inherently uncertain. For example, instruments cannot measure with perfect accuracy, samples are not exhaustive, variables change over time (in partially unpredictable ways), and abstractions and simplifications of the real world are necessary when resources are limited. While these imperfections are frequently ignored in GIS analyses, the importance of developing 'uncertainty aware' GIS has received increasing attention in recent years. Assessing and communicating uncertainty is important for establishing the value of data as an input to decision-making, for judging the credibility of decisions that are informed by data and for directing resources towards improving data quality. In this context, uncertainties in data propagate through GIS analyses and adversely affect decision making.

Error may be defined as the difference between reality and a representation of reality. In practice, errors are not exactly known. At best, users have some idea about the distribution of values that the error is likely to take. Perhaps it is known that the chances are equal that the error is positive or negative, or it may safely be assumed that in only one out of ten cases the absolute error is greater than a given number. For variables measured on a categorical scale, certain outcomes may be assumed more likely than others. Thus, uncertainty stems from an acknowledgement that errors may exist, and its magnitude is then proportional to how much is known about them. Uncertainty can be conveniently represented using probability theory, which provides a sound theoretical basis for assessing and propagating uncertainty. When probability distributions cannot be estimated, less detailed approximations may nevertheless be possible. These include intervals and scenarios, where possible outcomes are listed without their associated probabilities.

Historical Background

Interest in the problem of error and uncertainty in GIS dates back to the early years of GIS. For example, Peter Burrough devoted a chapter to the subject in his 1986 book on GIS. Burrough presented an exhaustive list of possible sources of error in spatial data and analysis. He also demonstrated the use of simple arithmetical rules from error analysis theory to propagate errors through GIS operations. Shortly after, the National Center for Geographic Information and Analysis in the USA defined uncertainty in spatial databases as one of its key research themes. The proceedings of a meeting on the topic were published in a book by Goodchild and Gopal in 1989. In the same year, Heuvelink and colleagues published an article in the International Journal of GIS that presented a more thorough statistical theory of spatial uncertainty propagation. These early publications, and the concurrent work on classification errors in remote sensing data, prompted many more studies on spatial uncertainty.

Since these early years, spatial uncertainty analysis has been a central theme in GI science. For example, three of the ten most cited papers in the first ten years of the International Journal of GIS address issues of data quality and error (Fisher 2001). Since 1994, conferences on spatial data quality or spatial accuracy assessment have been organised at least biannually. Statistical models characterizing uncertainty in spatial environmental variables have matured and can now reasonably describe both positional and attribute uncertainties. In recent years, greater emphasis has been placed on the communication of uncertainties and on the development of GI systems that can assess, store and simulate uncertain environmental data (i. e. emphasis has been placed on the practical application of uncertainty analysis).

Scientific Fundamentals

The description of uncertain spatial environmental variables benefits from a taxonomy of those variables. Here, an object-oriented approach is adopted, whereby 'spatial objects', such as houses, fields, rivers or cities, comprise one or more 'attributes', such as height, crop density, water quality or population size, respectively. Objects have boundaries that contain positional information, which may be uncertain. Attribute values may be defined at one or many locations for which the object is defined (e. g. water levels along a river) or as integral properties of the object (e. g. the length of a river). Attributes can be uncertain as well.

Positional Uncertainty

In order to describe the positional uncertainty of an environmental object, it is useful to classify objects by their primitive parts and by the types of movement they support under uncertainty:

- objects that are single points (point objects);
- objects that comprise multiple points whose internal geometry cannot change under uncertainty (rigid objects, lines as well as polygons);
- objects that comprise multiple points whose spatial relations can vary under uncertainty (deformable objects, lines as well as polygons).

The positional uncertainty of a point object always leads to a unitary shift (translation) in the object's position. This translation occurs in as many directions as the point has coordinate dimensions (e. g. *xyzt*). Unlike point objects, the positional uncertainty of a rigid object comprises a rotation of the object about an origin, as well as a simple translation. For a deformable object, the positional uncertainty of its individual points may lead to arbitrarily complex changes in the objects position or form. Such complexity occurs when the positional uncertainties of the individual points are partially or completely independent of each other (i. e. when the movement is more 'flexible').

Attribute Uncertainty

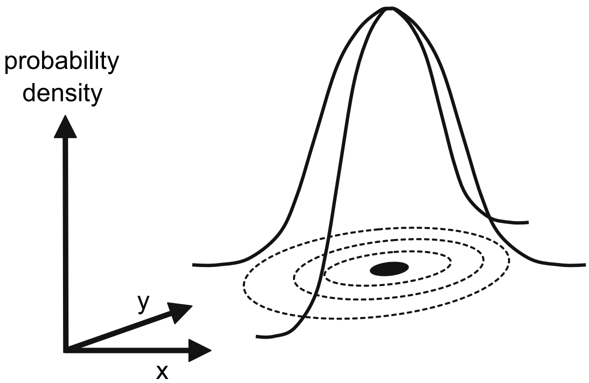
In order to characterise the uncertainty of a spatial attribute, it is useful to distinguish the value scale on which the attribute is measured. Three classes of measurement scale are distinguished, namely:

- attributes measured on a continuous numerical scale;
- attributes measured on a discrete numerical scale;
- attributes measured on a categorical scale.

Defining Probability Distributions for Positional Uncertainty

Methods for defining positional uncertainties in geographic objects include partial and full applications of probability theory to spatial vectors. They range in complexity from a simple 'epsilon (ε) band' approach, where a buffer of radius ε is imposed around each point or line segment, to the estimation of joint probability distribution functions (pdf) for the primitive nodes of the object. For a two-dimensional point object, this leads to a joint pdf of the x- and y-coordinates of a point, defined as:

$$F_{XY}(x, y) = \text{Prob}(X \leqslant x, Y \leqslant y)$$



Uncertain Environmental Variables in GIS, Figure 1 Sketch of bivariate probability distribution of a point object's position

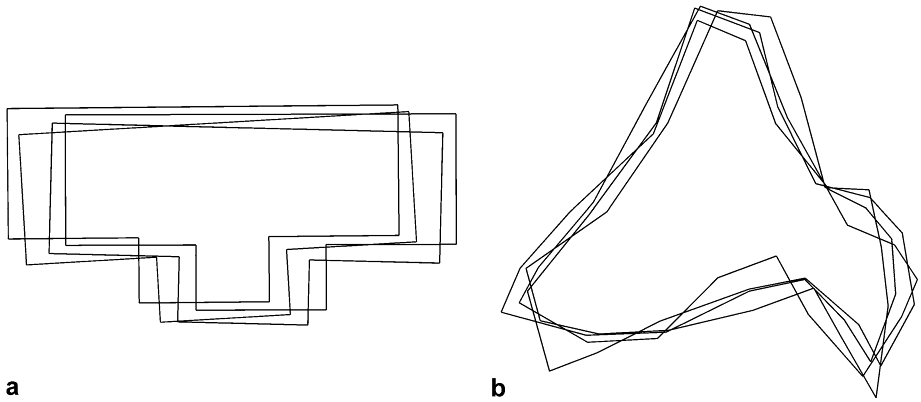
where X and Y are random variables representing the true, but unknown, position of the point. This is illustrated in Fig. 1. Important parameters of the pdf are the means and standard deviations in the x- and y-coordinates. The former determine the average position of the point in each direction and the latter determine its uncertainty.

For rigid objects that comprise multiple interconnected points, it is sufficient to define the translation and rotation of the object about a single reference point. Fig. 2a gives an example of four random draws of the position of an uncertain rigid object whose uncertainty in the x-direction is greater than that in the y-direction, and for which rotational uncertainty is relatively small.

For deformable objects, it is necessary to define the joint pdf of all the primitive points from which the object is comprised. Neighbouring points will rarely move independently because errors in measurement, georeferencing or cartographic generalization (among others) are often similar at nearby locations. This implies a need to model the relationships or correlations between the uncertain points. An example of a correlation model that depends on distance is the semivariogram. Fig. 2b gives an example where strong spatial correlation between neighbouring points is assumed, although the correlation is less than complete because the four realizations of the object are deformed.

Defining Probability Distributions for Attribute Uncertainty

Uncertainty in categorical attributes, as well as numerical attributes measured on a discrete scale, is represent-



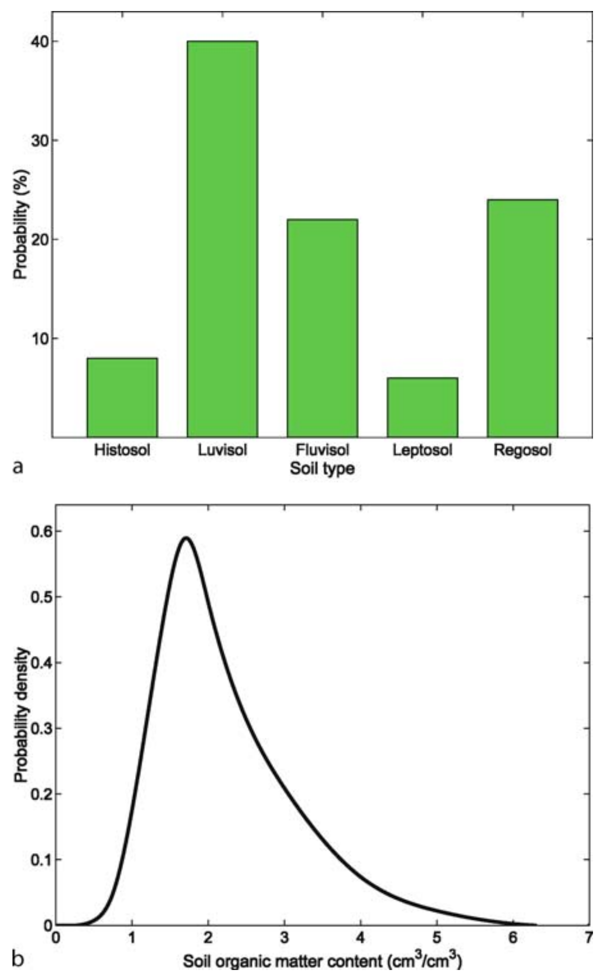
Uncertain Environmental Variables in GIS, Figure 2 Four random draws from the pdf of the position of an uncertain rigid object (**a**) and an uncertain deformable object (**b**)

ed with a discrete pdf. A discrete pdf simply comprises a probability for each possible outcome (Fig. 3a). For attributes measured on a continuous numerical scale, probability density functions are required, for which an example is given in Fig. 3b. For attributes that are distributed in space, a pdf is required for each location within the object, together with any spatial correlations between them. The normal (or Gaussian) distribution is a particularly common distribution for describing uncertain continuous numerical attributes, and is easily generalized to the spatial (multi-Gaussian) case. In contrast, it is usually much more difficult to incorporate spatial dependence in discrete numerical and categorical attributes.

Estimation of PDFs and Their Parameters

In order to apply one of the general pdfs for positional or attribute uncertainty to a particular case, the shape of the pdf and its associated parameters or class probabilities for categorical attributes must be known. Various approaches can be used:

- specification of instrument accuracy and precision (when variables are measured with instruments);
- kriging prediction variance (when variables are interpolated from point observations using geostatistical interpolation);
- classification error (when class maps are obtained with multivariate statistical methods such as maximum likelihood classification);
- goodness-of-fit measures such as the coefficient of determination, R^2 (when variables are derived from other variables through regression);
- calculation of root mean square error – RMSE (continuous variables) or confusion matrix (categorical variables) by comparison of the database values with ground truth or more accurate data sources;
- expert judgement.



Uncertain Environmental Variables in GIS, Figure 3 Discrete probability distribution for the categorical attribute soil type (**a**) and continuous probability distribution for the continuous numerical attribute soil organic matter content (**b**)

Despite the diversity of approaches, the identification and estimation of probability models is, in practice, the most difficult part of assessing uncertainty in environmental variables.

Uncertainty Propagation

When spatial calculations or operations rely on uncertain environmental inputs, these uncertainties will propagate through the calculations, leading to uncertain outputs. Since these outputs are often used in decision making, it is important to explore uncertainty propagation. For decisions that rely on a sequence of spatial calculations, uncertainty propagation must be examined throughout that sequence. There are several approaches for propagating uncertainty through spatial calculations. For example, when a linear calculation is performed on a normally distributed variable, the output will also be normally distributed, and will have a mean and variance that is analytically related to the mean and variance of the input. For non-linear calculations, the Taylor series method can be used to approximate the calculation with a linear or quadratic function, after which the propagation of uncertainty can also be computed analytically. The Taylor series method only applies when the spatial calculation is weakly non-linear and when all the uncertain inputs are continuous. In contrast, Monte Carlo simulation provides a general method for computing uncertainty propagation through spatial models. It requires the joint pdf of all the uncertain inputs, from which a random sample or 'realization' is easily obtained. By repeatedly sampling from the input pdf and running the model for each realization, an equivalently sized sample is obtained from the output pdf (i. e. the uncertain result of the spatial calculation). Thus, Monte Carlo Simulation (MCS) involves:

1. repeat N times (typically $100 \leq N \leq 10000$):
   a) generate a set of realizations from the uncertain input pdf using a pseudo-random number generator,
   b) for this set of realizations, compute and store the corresponding output,
2. compute and store the output for a subsequent calculation or compute sample statistics from the N outputs as an input to decision making (e. g. mean, standard deviation, percentiles).

The generality of MCS stems from the ease with which a random sample is obtained from a pdf and then used in a spatial calculation. However, its main weakness lies in the need to specify the input pdf (although recent advances are relaxing this requirement). For complex spatial calculations involving multiple uncertain inputs, any spatial correlations between these inputs are required to define the joint pdf. In addition, MCS is approximate because it samples from the output pdf (unlike analytical methods). Specifically, the accuracy of the method increases proportionally with the square root of the number of realizations and subsequent model runs, N (i. e. to double the accuracy one must quadruple the number of model runs). This gradual increase in accuracy implies that high accuracies are only obtained for large N. Thus, the method is relatively inefficient for complex spatial models (e. g. a distributed model of groundwater flow). However, there are several improvements over simple random sampling, which can dramatically improve the efficiency of MCS. As computers have become more powerful and models more complex (i. e. less tractable to analytical techniques), MCS has become increasingly popular in spatial modeling.

Key Applications

The assessment and propagation of uncertainty is a central theme in GI science, for which many references are provided in the 'recommended reading' section. A few of these are briefly outlined.

Positional Uncertainty

Shi and Liu (2000) present a stochastic model of the positional errors in line segments. The model assumes that the errors of the endpoints follow two-dimensional (x,y) normal distributions. Brown and Heuvelink (2007) present a software tool that can generate spatial objects with uncertain boundaries, where correlations between the primitive points of the objects are allowed. Strebelle (2002) uses stochastic imaging to explore the uncertainty in representing geological objects, such as sand channels and reservoirs, using training images.

Attribute Uncertainty, Continuous Data

Heuvelink (1998) describes how geostatistical techniques can be used to interpolate soil properties and to quantify the uncertainty in this interpolation. Van Oijen et al. (2005) use Bayesian calibration to estimate the uncertainty of continuous parameters of forest models. Fisher and Tate (2006) show how uncertainty in digital terrain models can be estimated from a comparison with ground truth observations.

Attribute Uncertainty, Categorical Data

Kros et al. (1999) present a model of uncertainty in soil type using indicator geostatistics. The parameters of the model are derived from a comparison of the large scale European soil map with a more detailed, and more accurate, local map for a case study site. Kyriakidis and Dungan (2001) also use a geostatistical approach to model the uncertainty in land cover from Landsat TM remote sensing imagery and point observations. Simulated land cover maps are then input into a biogeochemical model for assessing uncertainty in net primary production. Wickham et al. (2004) report how probability sampling was used to assess uncertainty in the US National Land Cover Data.

Uncertainty Propagation

Heuvelink (1998) uses the Taylor series method to assess uncertainty propagation through a multiple linear regression model. Miehle et al. (2006) analyze how errors in a GIS database propagate through the biogeochemical Forest-DNDC model and show that uncertainty in the predicted vegetation growth is strongly affected by upscaling. Karssenberg and De Jong (2005) present a modeling language which combines standard GIS scripts for environmental modelling with uncertainty propagation functionality. Crosetto and Tarantola (2001) show how the contribution of different sources of uncertainty to output uncertainty may be computed.

Future Directions

The design of pdfs that realistically describe the uncertainty about spatial environmental variables can benefit from the diverse set of stochastic spatial models that continue to appear in the scientific literature. For example, recent geostatistical models can make use of information derived from auxiliary variables (i. e. regression kriging), they can simulate complex spatial objects whose patterns and internal geometry must satisfy specific conditions (i. e. stochastic imaging), and can deal with categorical variables (i. e. indicator geostatistics; stochastic cellular automata; Bayesian methods).

Methods to estimate the parameters of pdfs can benefit from developments in expert elicitation and by an increased awareness among data providers that accuracy measures must be stored in spatial databases. Given that uncertainties typically originate from many different causes or 'sources', of which only some may contribute significantly to uncertainty propagation, a structured approach to evaluating the sources of uncertainty in data is important. In this context, tools that provide a conceptual framework for assessing uncertainty in environmental data are important, since they allow users without direct experience of uncertainty methods to develop realistic uncertainty models for their data (Brown and Heuvelink, 2007).

In the 1980s and 1990s, analytical approaches to uncertainty propagation (i. e. the Taylor series method) were popular. However, the simplicity and generality of the Monte Carlo method has secured its widespread application and longevity. The applicability of MCS has been further extended through improved sampling procedures. For example, Markov Chain Monte Carlo offers improved flexibility in simulating from complex pdfs, even when the pdf is not explicitly defined. MCS is also well suited to a parallel computing environment and will benefit from GRID computing in future.

Despite the great scientific interest in uncertainty, practical applications of uncertainty analysis remain uncommon in mainstream GIS. Although there are valid reasons for using uncertainty analysis in decision making, it would be naïve to claim that uncertainty analysis always pays off. A trade-off between the added value and the associated costs is always necessary. An uncertainty analysis often requires both considerable time and expertise, and cannot be reduced to pressing a single 'uncertainty analysis button'. Realistic assessments of uncertainty and the responsible use of uncertainty propagation analysis require that users have sufficient background in statistics. Educating GIS users in the theory and practice of uncertainty analysis is, therefore, essential. However, this must be accompanied by the development of user-friendly tools for performing uncertainty analysis and for communicating uncertainty to those who benefit from spatial calculations.

Cross References

► Error Propagation in Spatial Prediction
► Imprecision and Spatial Uncertainty
► Moving Object Uncertainty
► Positional Accuracy Improvement (PAI)
► Statistical Descriptions of Spatial Patterns
► Uncertainty, Modeling with Spatial and Temporal
► Uncertainty, Semantic

Recommended Reading

Arbia, G., Griffith, D.A., Haining, R.P.: Spatial error propagation when computing linear combinations of spectral bands: the case of vegetation indices. Environ. Ecol. Stat. **10**, 375–396 (2003)

Brown, J.D., Heuvelink, G.B.M.: The Data Uncertainty Engine (DUE): a software tool for assessing and simulating uncertain environmental variables. Comput. & Geosci. **33**, 172–190 (2007)

Crosetto, M., Tarantola, S.: Uncertainty and sensitivity analysis: tools for GIS-based model implementation. Int. J. Geogr. Inf. Sci. **15**, 415–437 (2001)

Ehlschlaeger, C.R., Shortridge, A.M., Goodchild, M.F.: Visualizing spatial data uncertainty using animation. Comput. & Geosci. **23**, 387–395 (1997)

Fisher, P.: Editorial, Citations to the International Journal of Geographical Information Systems and Science: the first 10 years. Int. J. Geogr. Inf. Sci. **15**, 1–6 (2001)

Fisher, P.E., Tate, N.J.: Causes and consequences of error in digital elevation models. Prog. Phys. Geogr. **30**, 467–489 (2006)

Foody, G.M., Atkinson, P.M. (eds.): Uncertainty in Remote Sensing and GIS. Wiley, Chicester (2002)

Goodchild, M.F., Guoqing, S., Shiren, Y.: Development and test of an error model for categorical data. Int. J. Geogr. Inf. Syst. **6**, 87–10 (1992)

Heuvelink, G.B.M.: Error Propagation in Environmental Modelling with GIS. Taylor & Francis, London (1998)
Heuvelink, G.B.M., Burrough, P.A. (eds.) Developments in Statistical Approaches to Spatial Uncertainty and its Propagation. Themed issue Int. J. Geogr. Inf. Sci. **16**, number 2 (2002)
International Spatial Accuracy Research Association. http://www.spatial-accuracy.org
Karssenberg, D., De Jong, K.: Dynamic environmental modelling in GIS. 2: Modelling error propagation. Int. J. Geogr. Inf. Sci. **19**, 623–637 (2005)
Kros, J., Pebesma, E.J., Reinds, G.J., Finke, P.F.: Uncertainty assessment in modelling soil acidification at the European scale: a case study. J. Environ. Qual. **28**, 366–377 (1999)
Kyriakidis, P.C., Dungan, J.L.: A geostatistical approach for mapping thematic classification accuracy and evaluating the impact of inaccurate spatial data on ecological model predictions. Environ. Ecol. Stat. **8**, 311–330 (2001)
Longley, P.A., Goodchild, M.F., Maguire, D.J., Rhind, D.W.: Geographic Information Systems and Science, 2nd edn. Wiley, New York (2005)
Miehle, P., Livesley, S.J., C.S. Li, Feikema, P.M., Adams, M.A., Arndt, S.K.: Quantifying uncertainty from large-scale model predictions of forest carbon dynamics. Global Change Biol. **12**, 1421–1434 (2006)
Shi, W., Liu, W.B.: A stochastic process-based model for the positional error of line segments in GIS. Int. J. Geogr. Inf. Sci. **14**, 51–66 (2000)
Shi, W., Fisher, P.F., Goodchild, M.F. (eds.): Spatial Data Quality. Taylor & Francis, London (2002)
Strebelle, S.: Conditional simulation of complex geological structures using multiple-point statistics. Math. Geo. **34**, 1–21 (2002)
Van Oijen, M., Rougier, J., Smith, R.: Bayesian calibration of process-based forest models: bridging the gap between models and data. Tree Physiol. **25**, 915–927 (2005)
Wickham, J.D., Stehman, S.V., Smith, J.H., Yang, L.: Thematic accuracy of the 1992 National Land-Cover Data for the western United States. R. Sens. Environ. **91**, 452–468 (2004)
Zhang, J.X., Goodchild, M.F.: Uncertainty in Geographic Information. Taylor & Francis, New York (2002)

Uncertainty

► Uncertainty, Modeling with Spatial and Temporal

Uncertainty, Modeling with Spatial and Temporal

GEORGE CHRISTAKOS
Department of Geography, San Diego State University, San Diego, CA, USA

Synonyms

Knowledge synthesis; Spatial; Temporal; Bayesian maximum entropy; Uncertainty; Interdisciplinary

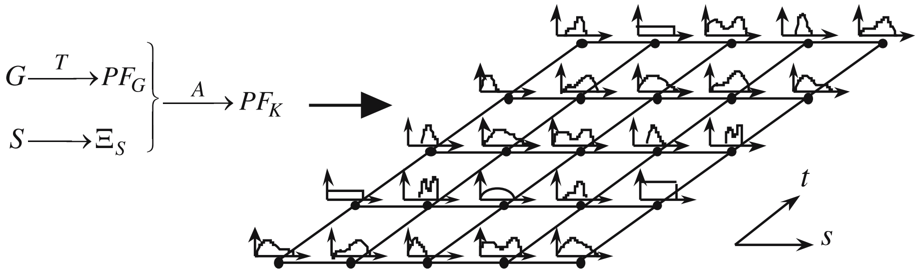
Definition

Bayesian maximum entropy (BME) [1] is a general approach used in the study of geographical systems and attribute fields (natural, social, cultural, etc.) that are characterized by space–time dependence and multisourced uncertainty. The BME modeling and mapping of attribute fields is a fruitful combination of: (1) a methodology offering the conceptual means to integrate the various mental entities of spatiotemporal analysis (theories, techniques and thinking modes), (2) a stochastic theory providing adequate representation of spatiotemporal dependence and multisourced uncertainty, and (3) a technology that provides the necessary means to process and visualize the conceptual, theoretical and methodological results of items 1 and 2 above. Figure 1 is a schematic representation of the BME approach [2]: T and A denote the teleologic and adaptation principles, respectively (the former operates on the general knowledge base, G-KB, whereas the latter involves the specificatory KB, S-KB); Ξ_S denotes a set of operators that express mathematically the S-KB; PF_G is a probability function derived from G-KB by teleologic means, and PF_K is a probability function that updates PF_G in view of S and A. Thus, the PF_K distribution across space–time accounts for the total KB, $K = G \cup S$. From PF_K, several kinds of substantive attribute maps can be derived: maps of field predictions, uncertainty assessments and simulations across space–time. For example, if the attribute of interest is population mortality across a region, BME synthesizes the KBs available concerning the situation and generates a stochastically complete representation in terms of maps of the resulting PF_K at any geographical location and time period of interest; from PF_K, mortality predictions are derived together with the associated accuracy across space–time, risk estimates for specified population cohorts are made, etc.

Historical Background

U

Space–time or spatiotemporal data analysis in a modern statistical framework was introduced in [3,4,5]. Subsequent works include [6,7,8]; see relevant literature for a detailed list of publications on the spatiotemporal statistics subject. The BME conceptual framework and quantitative techniques of spatiotemporal modeling and mapping under conditions of multisourced uncertainty were originally introduced as the powerful combination of Boltzmann–Shannon information concepts and operational Bayes theory in a composite space–time environment [3,9]. Later, the BME was considered in the general context of epistematics, which is the fusion of human teleology and evolutionary epistemology [10]. Naturally, BME developments followed a variety of paths, depending



Uncertainty, Modeling with Spatial and Temporal, Figure 1 A schematic representation of the main Bayesian maximum entropy (BME) steps across space (*s*) and time (*t*). *T* and *A* denote the teleologic and adaptation principles, respectively (the former operates on the general knowledge base, *G*-KB, whereas the latter involves the specificatory KB, *S*-KB), Ξ_S denotes a set of operators that express mathematically the *S*-KB; PF_G is a probability function derived from *G*-KB by teleologic means, and PF_K is a probability function that updates PF_G in view of *S* and *A*

on the scientific discipline considered, and the BME techniques have been successfully used in real-world studies in physical and medical geography, earth and atmospheric sciences, environmental engineering, epidemiology, health science, human exposure, risk assessment, and decision analysis [11,12,13,14,15,16,17,18,19].

The term "Bayesian" in BME denotes that multisourced site-specific KBs are assimilated using an operational yet physically meaningful space–time Bayesian principle (there are salient distinctions between this principle and the standard Bayesian rule of classical statistics [10]). The term "maximum entropy" implies that general (core) knowledge is integrated in a teleology of reason manner that assures the generation of a maximally informative stochastic model of the natural system. The need for an epistemologically sound methodology stemmed from the fact that, even if the world is ontologically unitary, our knowledge of it is epistemologically diverse (due to technological, cultural and historical constrains).

Scientific Fundamentals

BME distinguishes between general (core), *G*-KB, and specificatory, *S*-KB. G-KB may incorporate scientific theories, physical laws, biologic principles, mechanistic relations, ecologic functions, social structures, population dynamics, etc.; *S*-KB accounts for hard data, soft information, secondary sources, uncertain observations, categorical data, fuzzy inputs, etc. [19]. Thus, G-KB comprises more of the intellectual side of education and research, with more emphasis on the cognitive and theoretical, whereas *S*-KB comprises the more experiential, subjective, intuitively apprehended side. A main BME objective is to integrate various forms of core and site-specific KB in order to generate informative maps and meaningful statements. While some of the KBs include rigorous quantitative assessments, many others are about multisourced beliefs about the situation of interest. In many cases, the KBs do not refer directly to the beliefs but rather to the sentences that people use to state them. There is certain amount of hazard in this, which is yet another good reason for using stochastic concepts and techniques.

BME involves a state-of-the-art stochastic theory [5]. There is a good reason for this: in real-world applications, uncertainty is a major factor expressed either in terms of statistical probabilities (related to ontologic features of the actual system) or by means of inductive probabilities (representing modeler-dependent considerations); the meaning of space–time metrics or distances may depend on the physical or social conditions of the situation; and it is often the observation scale that determines our conceptualization of the phenomenon. In the context of geographical information science, stochastic theory is a considerable advancement over mainstream statistics, since it does not suffer from the limitations of the latter such as: classical statistics uses the independence assumption (attributes are modeled in terms of independently distributed random variables); many statistical tests entail logical problems and are often irrelevant to the objectives of a physical study (e. g., a statistical test states the probability of the observed outcome given that a null hypothesis is true, whereas a scientific investigation often seeks the probability that the null hypothesis is true given the observed outcome); in spatial statistics the physics of space and time are not adequately taken into consideration; and spatial econometrics may be lacking rigorous mechanisms to incorporate various important forms of core and specificatory knowledge.

In light of the above considerations, BME can evaluate distinct uncertainty types (conceptual and technical, ontologic and epistemologic); its scientific background involves teleologic (in the human teleology sense) and adaptation (in the evolutionary sense) principles, which can embrace diverse phenomena and interdisciplinary descriptions in a single scheme; it assumes space–time coordinate systems that accommodate attribute variability; it represents space–time dependence patterns in terms of stochastic spatiotemporal models (covariances and variograms, separable and nonseparable, ordinary and generalized [1,2,3,4,5,10]; it offers complete system characterization in terms of prediction probability laws—not necessarily Gaussian—at every grid point (and not merely a single prediction at each point); it automatically incorporates nonlinear attribute predictors (rather than restrictive linear or linearized field estimators); it relies on a natural knowledge-based methodology (rather than on mechanistic curve fitting, ad hoc trend surface, etc. techniques); it provides operational Bayes assimilation rules that are effective and considerably flexible; and it can handle different kinds of space–time support (functional BME) as well as more than one attribute (vector BME or co-BME) [1]. Many of the previous results (e. g., spatial regression and geostatistical kriging techniques) are easily derived as special cases of BME under limited conditions.

Implementation of the BME concepts and techniques requires the development of adequate software packages of spatiotemporal analysis and mapping. Such a package is the Epistematics Knowledge Synthesis and Graphical User Interface (SEKS-GUI) software library, which comprehensively features [20]: (1) the theoretical support of interdisciplinary knowledge synthesis and random field mathematics, (2) a user-friendly interface for space–time modeling and mapping through a series of screens, (3) built-in functions for intermediate steps so that users need not handle individual library functions nor connect the software processing stages; (4) a complete graphics-based environment that offers significant flexibility in providing the input, deciding the investigation course, choosing among an array of available attribute predictions and selecting from a broad variety of output visualization options. Mapping does not merely refer to the geographical distribution of attributes, which means that a meaningful visualization must take into consideration the background, expertise and objectives of the user of the maps. SEKS-GUI's visualization options include the following: maps of the complete probability functions and predictions maps (in the form of mean, mode etc.) are derived at each space–time point; and accuracy measures (such as prediction errors, intervals and sets) are also available.

Key Applications

BME models and techniques are used in a variety of application domains (scientific, cultural, social, etc.) A few of these application domains are reviewed below (for more details and additional applications domains, the interested reader should consult the relevant literature).

Earth and Atmospheric Sciences

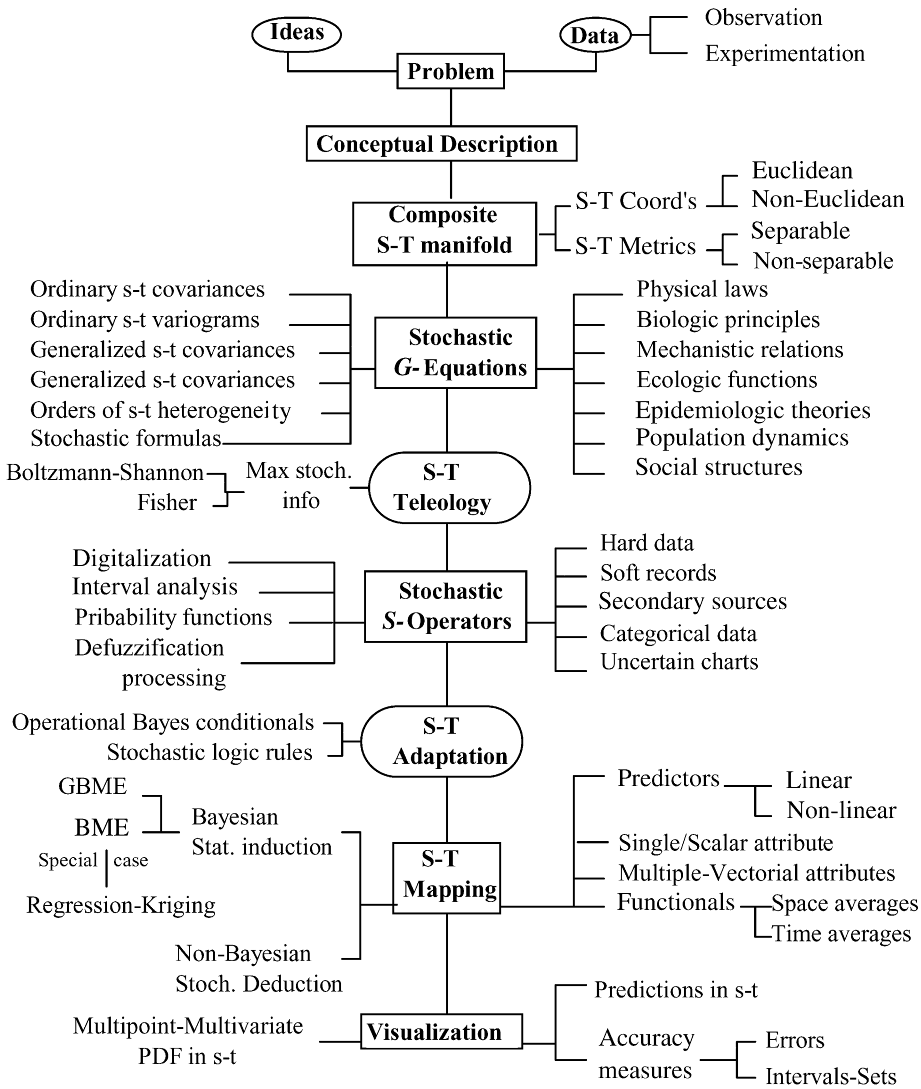
BME analysis has been implemented in total ozone mapping by integrating datasets from remote sensing instruments (satellite-based Total Ozone Mapping Spectrometer (TOMS) and Solar Backscatter Ultraviolet(SBUV) measurements) and empirical models (USA); in powering an air quality information system in Cairo (Egypt); in geophysical assimilation of several forms and under varying conditions; in the analysis of springwater ion processes derived from measurements at the Dyle Basin (Belgium); in the hydrogeologic modeling of the Equus Beds aquifer (Kansas); in the estimation of horizontal hydraulic conductivity for the Kirkwood–Cohansey aquifer (New Jersey); in the incorporation of physical laws (Darcy, advection-reaction law etc.) to predict hydrologic variables and chemical concentrations; in porous media upscaling studies; in continuous-valued map reconstruction; in the space–time mapping of soil salinity; in the solution of partial differential equations representing the geothermal field in Nea Kessani (Greece); in soil property estimation from thematic soil maps; in the calibration/validation of an altimeter wave period model and the application to TOPEX/Poseidon (Topography Experiment) and Jason-1 altimeters; in the space–time representation of pollutant distributions at various geographical regions of the world; and in the analysis of radioactive soil contamination [5,12,13,14,15,19,21,22,23,24,25].

Geography, Health and Epidemiology

BME techniques have been used to study the geographical-temporal patterns and propagation dynamics of major epidemics (e. g., the Black Death epidemic in fourteenth century Europe and bubonic plague in late nineteenth to early twentieth century India); to combine categorical and continuous information; to asses influenza risk and the effects of El Niño weather conditions in the state of California; to study the multiscale features of space–time mortality distributions; to perform geographical management and decision analysis in multidimensional environments; and to analyze and map sexually transmitted diseases to optimize intervention and prevention strategies [11,16,26,27].

Human Exposure

BME has been used in the mapping of contaminant exposure in south-west Russia due to the Chernobyl fallout in Ukraine; in the assessment of health effects at local/global scales due to various attributes (ozone, particulate matter distribution, sulfate deposition, etc. concentrations) and for different population cohorts; in the mapping of arsenic concentrations in Bangladesh drinking water and the prediction of population health effects; in the representation of



Uncertainty, Modeling with Spatial and Temporal, Figure 2 Outline of interdisciplinary knowledge synthesis (IKS) (methodology, stochastic modeling, real-world implementation). *S–T* Space–time, *GBME* generalized *BME*, *PDF* probability density function

temperature-mortality causal associations; in the study of health effects of air pollution on lupus (North and South Carolina); and in the determination of space–time lead contamination at the Cherry Point Air Force site (North Carolina) and the corresponding indicators of effects on children's arithmetic abilities [12,17,28,29].

Future Directions

The BME techniques constitute an important component of interdisciplinary knowledge synthesis (IKS) [10]. IKS institutes a broad framework in which different sets of mental entities describing constituent phenomena in the individual disciplines are integrated to solve (describe, explain and predict) composite real-world problems. Figure 2 briefly outlines the various IKS elements. Naturally, the starting point is the generation of an important problem, as mentioned above. The IKS methodology entails various forms of mathematical expressions: stochastic G-equations, S-operators, information measures, Bayes conditionals and stochastic logic rules. In fact, the introduction of mathematical expressions in the geographical reasoning context seeks the unity of subject (modeler) and object (geographical system) in an operational framework that provides the means for rigorous quantitative assessments (generating predictions, assessing space–time dependence, characterizing uncertainty etc.). IKS establishes a very general framework for integration of interdisciplinary KB (scientific, cultural, social, economic etc.) and argumentation modes (taxonomic, analogical, mathematical, experimental, etc.). This framework possesses explanatory and predictive context, in addition to descriptive. In the IKS context, the list of potential BME applications is endless [2,10,30]. IKS plays a vital role in the emerging Conceptual Age. This being the case, IKS would face a number of significant challenges, as follows:

- How are the composite attributes of a system as a whole related to the constituent attributes? In some cases, it is possible to apply a kind of an isolation condition: the composite attributes of the structured whole are, in a sense, mirror images of the constituent attributes; their behavior in the structured whole can be then derived from constituent attributes plus statements describing the organized structure in which they are bound and the prevailing system conditions. In many other cases, however, a connection condition applies: it is impossible to understand how the composite attributes function by simply studying the corresponding constituent attributes in isolation conditions.
- How can temporal geographical informations systems (GIS) account for differences having to do with the way each scientific discipline communicates knowledge? Physical sciences use mainly mathematical formulas and models to express conceptual, observational and experimental results. In humanistic disciplines, there is little resort to mathematical formulas: chiefly, reliance is placed upon analogy and metaphor.
- What is the nature of space–time? The matter is metaphorically expressed by the "canvas versus parenthood dilemma": is space–time like a canvas that exists whether or not the artist paints on it, or is space–time akin to parenthood that does not exist until there are parents and children? A related question is the "asymmetry of time": is time's asymmetry a property of states of the world rather than a property of time as such?
- What knowledge bases are most reliable and/or important? To address oneself to this question is to ask for a classification of sorts of knowing, a ranking of these sorts by reference to some reliability/value standards, and a meaningful uncertainty characterization (conceptual versus technical, etc.).

Cross References

► Data Analysis, Spatial
► Public Health and Spatial Modeling
► Spatial Uncertainty in Medical Geography: A Geostatistical Perspective
► Statistical Descriptions of Spatial Patterns

Recommended Reading

1. Christakos, G.: Modern Spatiotemporal Geostatistics. Oxford University Press, New York (2000)
2. Christakos, G., Bogaert, P., Serre, M.L.: Temporal GIS. Springer-Verlag, New York (2002)
3. Christakos, G.: Random field modelling and its applications in stochastic data processing. Ph.D. Thesis, Harvard University (1990)
4. Christakos, G.: On certain classes of spatiotemporal random fields with application to space-time data processing. IEEE Trans. Syst. Manuf. Cybernetics **21**, 861–875, (1991)
5. Christakos, G.: Random Field Models in Earth Sciences. Academic, San Diego (1992)
6. Goodall, C., Mardia, K.V.: Challenges in multivariate spatio-temporal modeling. In: Proceedings of the XVIIth International Biometric Conference, Hamilton, Ontario, Canada, 8–12 August 1994
7. Bogaert, P.: Comparison of kriging techniques in a space-time context. Math. Geol. **28**, 73–86 (1996)
8. Kyriakidis, P.C., Journel, A.G.: Geostatistical space-time models: a review. Math. Geol. **31**, 651–684 (1999)
9. Christakos, G.: A Bayesian/maximum-entropy view to the spatial estimation problem. Math. Geol. **22**, 763–776 (1990)
10. Christakos, G., Olea, R.A., Serre, M.L., Yu, H.L., Wang, L.-L.: Interdisciplinary Public Health Reasoning and Epidemic Modelling: The Case of Black Death. Springer-Verlag, New York (2005)
11. Serre, M.L., Christakos, G.: Modern Geostatistics: Computational BME in the light of uncertain physical knowledge–The Equus Beds study. Stochastic Environ. Res. Risk Assess. **13**, 1–26 (1999)
12. Serre, M.L., Kolovos, A., Christakos, G., Modis, K.: An application of the holistochastic human exposure methodology to naturally occurring arsenic in Bangladesh drinking water. Risk Anal. **23**, 515–528 (2003)
13. Bogaert, P., D'Or, D.: Estimating soil properties from thematic soil maps-The BME approach. Soil Sci. Soc. Am. J. **66**, 1492–1500 (2002)
14. Kolovos, A., Christakos, G., Serre, M.L., Miller, C.T.: Computational BME solution of a stochastic advection-reaction equation in the light of site-specific information. Water Res. Res. **38**, 1318–1334 (2002)
15. Papantonopoulos, G., Modis, K.: A BME solution of the stochastic three-dimensional Laplace equation representing a geothermal field subject to site-specific information. J. Stochastic Environ. Res. Risk Assess. **20**, 23–32 (2006)
16. K-M Choi, Christakos, G., Wilson, M.L.: El Niño effects on influenza mortality risks in the state of California. J. Public Health **120**, 505–516 (2006)
17. Savelieva, E., Demyanov, V., Kanevski, M., Serre, M.L., Christakos, G.: BME-based uncertainty assessment of the Chernobyl fallout. Geoderma **128**, 312–324 (2005)
18. Christakos, G., Li, X.: Bayesian maximum entropy analysis and mapping: A farewell to kriging estimators? Math. Geol. **30**, 435–462 (1998)
19. Bogaert, P.: Spatial prediction of categorical variables: the BME approach. Stochastic Environ. Res. Risk Assess. **16**, 425–448 (2002)
20. H-L Yu, Kolovos, A., Christakos, G., J-C Chen, Warmerdam, S., B. Dev. Interactive spatiotemporal modelling of health systems: The SEKS-GUI framework. In: Griffith, D.A., Christakos, G. (eds.) J. SERRA–Special Issue on Medical Geography as a Science of Interdisciplinary Knowledge Synthesis under Conditions of Uncertainty (2007, in press)
21. Douaik, A., van Meirvenne, M., Toth, T., Serre, M.L.: Space-time mapping of soil salinity using probabilistic BME. J Stochastic Environ. Res. Risk Assess. **18**, 219–227 (2004)

22. Quilfen, Y., Chapron, B., Collard, F., M.L Serre.: Calibration/validation of an altimeter wave period model and application to TOPEX/Poseidon and Jason-1 Altimeters. Mar. Geodesy **3/4**, 27, 535–550 (2004)
23. Serre, M.L., Christakos, G., Howes, J., Abdel-Rehiem, A.G.: Powering an Egyptian air quality information system with the BME space/time analysis toolbox: Results from the Cairo baseline year study. In: Monestiez, P., Allard, D., Froidevaux, R. (eds.) Geostatistics for Environmental Applications, pp. 91–100. Kluwer, Dordrecht (2001)
24. Vyas, V.M., Tong, S.N., Uchrin, C., Georgopoulos, P.G., Carter, G.P.: Geostatistical estimation of horizontal hydraulic conductivity for the Kirkwood-Cohansey aquifer. J. Am. Water Res. Assoc. **40**, 187–195 (2004)
25. Christakos, G., Kolovos, A., Serre M.L., Vukovich, F.: Total ozone mapping by integrating data bases from remote sensing instruments and empirical models. IEEE Trans. Geosci. Remote Sensing **42**, 991–1008 (2004)
26. H-L Yu, Christakos, G.: Spatiotemporal modelling and mapping of the bubonic plague epidemic in India. Int.. J. Health Geogr. **5** (2006) Available via: http://www.ij-healthgeographics.com/content/5/1/12
27. Gesink-Law, D.C., Bernstein, K., Serre, M.L., Schumacher, C.M., Leone, P.A., Zenilman, J.M., Miller, W.C., Rompalo, A.M.: Modeling a syphilis outbreak through space and time using the BME approach. Ann. Epidemiol. **16**, 797–804 (2006)
28. Yu, H-L, Chen, J.-C., Christakos, G., Jerrett, M.: Estimating residential level ambient PM10 and Ozone exposures at multiple time-scales in the Carolinas with the BME method. In: Internal Report IR09, Geography Department, San Diego State University, San Diego, CA (2007)
29. Augustinraj, R.: A Study of Spatiotemporal Health Effects due to Water Lead Contamination. M.S. Thesis, University of North Carolina (2002)
30. Christakos, G.: On a deductive logic-based spatiotemporal random field theory. Probability Theor. Math. Stat. (Teoriya Imovirnostey ta Matematychna Statystyka) **66**, 54–65 (2002)

Uncertainty, Semantic

PETER FISHER
Department of Information Science, City University, London, UK

Synonyms

Ambiguity; Discord or non-specificity in spatial data; Senses, alternative; Meaning, multiple; Folksonomy; Vagueness; Non-specificity; Semantic discord

Definition

When two people look at an instance of the same phenomenon and use a different vocabulary or use the same words in different ways, semantic uncertainty arises. Unfortunately in the storage, manipulation and querying of geographical information this happens very frequently. To organize and assist understanding their environment humans seem to need to identify and name phenomena, often making arbitrary and vague partitions from a continuum of observable properties of the phenomenon. The naming of the phenomenon however enables people to make reference to features and phenomena so that they can communicate ideas, and opinions about those phenomena to others, and to use them as referents in describing locations and landscapes. In the geographical sciences it enables us to map those phenomena by placing lines or text on a map, or in a database. Semantic uncertainty therefore arises in many situations. Most obviously it is a problem when translating between different languages, but it is frequently a problem within any one language, as well.

Historical Background

In some respects the study of semantic uncertainty is in its infancy within geographical information science, although the issue has always been present for as long as people have named geographical phenomena and features. In technical subject domains, which now use geographical information technology, it has been a concern for as long as people have been using those systems.

Consider soil mapping, for example, which is a frequently used input to geographical information system applications. Soil mapping involves the delineation of areas of uniform soil properties usually known as soil series or phases. The soil within one series is part of a hierarchical system of classification, but the details of the hierarchy vary by country and mapping organization. Some classification schemes are intended for global application, but the creation of the *FAO/UNESCO Soil Map of the World* published between 1971 and 1981 from national soil maps was a matter of considerable effort, necessitating the alignment of classifications. Furthermore, any soil mapping scheme reflects the "best" knowledge of the scientific discipline at any time, but as that knowledge changes or improves over time the classification scheme necessarily changes. Some changes are minor but others can be more fundamental. Soil classification, for example, experienced a major change with international repercussions through the publication of the US Soil Taxonomy, as a fundamental revision of the 1938 US soil classification.

Similar changes and problems can be seen in many areas of classifications of geographical information. The understanding of the geological stratigraphy of a region may change and result in a revision of how rocks are expected to occur in the stratigraphic sequence and how that sequence is expressed in the geological map. Changes of this type have resulted in the remapping of areas when the rocks have not changed.

Since the advent of satellite remote sensing, there have been numerous international and national mapping programmes of land cover. In spite of the influence of the USGS Land Use and Cover Classification, each mapping exercise tends to have used a different classification scheme.

These all result in the discordant use of technical language: use of the same word to mean different types of things, or different words to refer to the same type of thing. Historically distinguishing these classes has been accompanied by long text descriptions, and extensive annotation in published reports (known variously as bulletins, survey reports, memoirs etc.). The objective of these documents is to explain the classification to interested readers and the differences between properties.

The naming by people of features in the landscape also gives rise to uncertainty both as to the existence of those features and their extent. The presence of a mountain, for example, is something on which many observers may agree, but the actual extent of the mountain is harder to define. Even if one person may be definite about the extent, their opinion may be different from anyone else's or even their own on another occasion. The description of many features in the human landscape including, for example, villages, towns and cities, is associated with such uncertainty.

Natural language is even more a source of uncertainty than formal language. Less technical forms of semantics are harder to document. Semantic uncertainty is also concerned with the social naming of features, including that of inhabitated places.

This is the concern of recent research in ontology and folksonomy. Ontology is structured around the formal definitions of terms and the associations between terms and classes of things, and is the subject of another entry in this Encyclopedia. Folksonomies are used to classify concepts associated with information retrieval from the internet, and involve approximating at ontology from how people use language.

Scientific Fundamentals

Three principal approaches to semantic uncertainty can be identified as being relevant to spatial information, vagueness, non-specificity and discord (the last two being collectively referred to as ambiguity) [3]:

Vagueness

Vagueness occurs when the identification of the presence of a phenomenon is a matter of degree. Generally possessing a property (or properties) or a threshold amount of a property (or properties) can be used to distinguish a location as belonging to a set or class. However, in many instances the threshold value of the property is not a definite cutoff, and the membership of the set is a matter of degree; if more than a certain amount is present the location is definitely in the set, if less than a smaller amount is present it is definitely not in the set, but if in intermediate amount is present then it belongs to the set to a degree. This type of uncertainty is associated with the use of language but is very well treated by fuzzy set theory, which is the subject of a separate entry in this Encyclopedia.

Non-specificity

Non-specificity arises when the boundary condition that allows the identification of a phenomenon as being present is not well defined. It is possible that this is due to having a well defined algorithm for detecting the condition of the phenomenon but no agreement over the parameters of that algorithm, or it may be simply that the condition is diagnosed from a threshold and there is no agreement on the threshold value. The result is that people may identify different locations with the same named condition, and even if they agree on the naming of a location, the extent over which they consider the naming relevant may not be agreed.

Various researchers have used multiple realizations (or precisifications) of the identification of the phenomenon to generate fuzzy and probabilistic extents of the phenomenon [4,7]. Multiple levels of vagueness have been identified associated with multiple parametric uncertainties [5].

Discord

Discord occurs when completely different naming conventions are used. In rare occasions similar thresholds are used but the naming is different and there can be a one-to-one relationship between classes in alternative classifications. More normally there is a many-to-many relationship between classes.

Methods for working with these situations are in their infancy. Most methods involve using a weighted cross-tabulation of the two classification schemes. The tabulation is populated in different ways in the various studies. One study has used fuzzy numbers from experts describing the degree to which classes are similar [6]. Another has used fuzzy memberships of the boundary conditions between classes to populate the matrix [1]. A third approach has used two lines of evidence including automated extraction from object-based metadata and expert opinion [2]. All methods show some degree of success in the aim of merging discordant semantic classifications, but none is yet identifiable as the preferred method.

Key Applications

Semantic uncertainty arises in situations where humans give names to things, and especially when those things are mapped as geographical information. It therefore occurs whenever two analysts do not have an absolutely clear and identical conceptualization of the classes of objects they are studying. It is especially a problem with existing databases, being compared with other existing datasets or with new datasets.

Particular situations in which semantic uncertainty arises is when a single theme is being mapped in separate jurisdictions as independent exercises and people wish to merge the results of those surveys. So when multi-agency, multi-state or multi-national mappings of phenomena (land cover, soil, geology, vegetation, cadastral, etc.) are created the likelihood is that object naming one side of the boundary between jurisdictions does not match the naming the other side of the boundary. Indeed a well recognized problem is that of boundary matching across the boundary (so-called zipping).

The issue also arises when multiple datasets exist for the same area at different times. Frequently, the naming convention at different times is different for many different reasons and there is no guarantee that the instances of objects at the non-spatial boundary between classes are treated identically.

Future Directions

Current research in ontologies and the semantic web promises to address the issue of semantic uncertainty, but very little of that work is really addressing the matching of legacy databases. Boundary conditions and namings across boundaries may also be taken further by the work in fuzzy and rough sets. Text mining is showing some promise for populating the comparison matrices, but this work is still in its infancy.

Cross References

► Ontology-Based Geospatial Data Integration

Recommended Reading

1. Ahlqvist, O.: Using uncertain conceptual spaces to translate between land cover categories. Int. J. Geogr. Inf. Sci. **19**, 831–85 (2005)
2. Comber, A., Fisher, P., Wadsworth, R.: Integrating land cover data with different ontologies: identifying change from inconsistency. Int. J. Geogr. Inf. Sci. **18**, 691–708 (2004)
3. Fisher, P.F.: Models of Uncertainty in Spatial Data. In: Longley, P., Goodchild, M., Maguire, D., Rhind, D. (eds.) Geographical Information Systems: Principles Techniques Management and Applications, vol. 1, pp. 191–205. Wiley and Sons, New York (1999)
4. Fisher, P.F., Wood, J., Cheng, T.: Where is Helvellyn? Multiscale morphometry and the mountains of the English Lake District. Trans. Inst. British Geogr. **29**, 106–128 (2004)
5. Fisher, P.F., Cheng, T., Wood, J.: Higher order vagueness in geographical information: Empirical geographical population of Type n fuzzy sets. GeoInformatica (in press)
6. Fritz, S., See, L.: Comparison of land cover maps using fuzzy agreement. Int. J. Geogr. Inf. Sci. **19**, 787–807 (2005)
7. Montello, D.R., Goodchild, M.F., Gottsegen, J., Fohl, P.: Where's downtown?: Behavioural methods for determining referents of vague spatial queries. Spatial Cogni. Comput. **3**, 185–204 (2003)

Unified Index Scheme

► Geospatial Authorizations, Efficient Enforcement

Unified Modeling Language Extension Mechanisms

JUGURTA LISBOA FILHO, JOSÉ LUÍS BRAGA
Department of Computer Science/Information Systems, Federal University of Vicosa (UFV), Vicosa, Brazil

Synonyms

Stereotypes

Definition

The Unified Modeling Language (UML) provides mechanisms that enable the extension of its own constructors, i. e., new items can be defined and integrated into the language, without having to modify the underlying modeling language. Extensions can be made to adapt the language to specific purposes such as to tailor the UML metamodel for different platforms (e. g., Java2 Platform Enterprise Editions, J2EE, or .NET Framework) or to support the specification of own/particular restrictions of a certain domain [e. g., geographic information systems (GIS) applications]. The UML metamodel consists of the formal description of the language itself and is established by Object Management Group (OMG) [3]. The UML extensibility mechanisms include profiles, constraints, tagged values, and stereotypes.

Main Text

UML is a naturally expandable language. The great variety of application domains makes it practically impossible to define a single model that appropriately assists the more specific modeling requirements of each domain. Therefore, when the UML was created, it was already clear that its model constructors would not be enough and because of this some extension mechanisms were introduced into the language [1].

The first UML version [2] introduced the concepts of stereotype and tagged value, which were extensions based on strings that could be added to the items of the UML diagrams in a very flexible way. Afterwards, the Profile concept was defined in order to provide more structure and precision to the definition of stereotypes and tagged value, for a particular domain [3].

Profiles enable existent metamodels to be extended, i. e., that new specific metaclasses of a domain are defined and incorporated into the other elements of the model, constituting new constructors that become part of the UML. A classic example is the UML Profile Business Modeling, which describes how the UML can be customized for modeling business applications. This profile defines a set of stereotypes composing a useful terminology that can be incorporated into the classes of a business application. Another example is the UML-GeoFrame Profile, described in this encyclopedia, which extends the UML to modeling spatiotemporal databases.

Cross References

► Modeling with a UML Profile
► UML: Unified Modeling Language

Recommended Reading

1. Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language Reference Manual. Addison-Wesley, Reading, MA (1999)
2. Object Management Group.: OMG Unified Modeling Language Specification, Version 1.5. Object Management Group, Needham, MA (2003)
3. Object Management Group.: Unified Modeling Language: Superstructure, Version 2.0,.Object Management Group, Needham, MA (2005)

Unified Modeling Language-Geoframe Modeling Language

► Modeling with a UML Profile

University of Minnesota (UMN) Map Server

RANGA RAJU VATSAVAI[1], THOMAS E. BURK[2], STEVE LIME[3]
[1] Computational Sciences and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN, USA
[2] Department of Forest Resources, University of Minnesota, St. Paul, MN, USA
[3] Minnesota Department of Natural Resources, Minneapolis, MN, USA

Synonyms

MapServ; UMN MapServer; WebGIS

Definition

Recent efforts on exploiting the Web to disseminate geographic information to the general public using the Internet has resulted in a new breed of specialized geographic information systems, generally called WebGISes or mapservers. A mapserver can broadly be defined as a software platform for dynamically generating spatially referenced (digital) map products from geo-spatial data. It is an integrated tool for building and serving WebGIS applications. The University of Minnesota MapServer or UMN-MapServer, or simply MapServer, is one such system. It was originally developed as part of a research project at the University of Minnesota. Although it is still supported and distributed through the University of Minnesota, it has evolved into an international Open Source project, utilizing Open Source GIS libraries developed by experts from around the world. MapServer is not only an immensely popular Open source development environment for building spatially-enabled Internet applications, but it is also a melting pot for experimenting with several innovative and advanced features. Typical functionality of Mapserver includes query and visualization. However, systems such as MapServer provide additional features like advanced cartographic output, scripting interfaces, on-the-fly map projection, spatial analysis and editing.

A typical Mapserver system consists of two major components, the client and the server. Based on how the functionality is shared among the client and server components, a mapserver can be classified as a client-centric or a server-centric system. With a server-centric approach, the server is responsible for everything, starting with data access, query processing and analysis, to final output (typically an image map) generation. On the other hand, with client-centric approaches, the server is typically responsible only for data access. Processing is handled by the client. However, with recent advances in hardware and software (programming) environments, hybrid approaches where functionality can be selectively distributed between the client and server are becoming popular. Traditional mapservers follow a server-centric approach. An extended version of MapServer [13] follows an innovative architecture where any given geo-spatial analysis function is executed either on the server or the client, based on some performance criteria. This hybrid architecture has shown to improve the system performance and allow support of online geospatial analysis.

Basic MapServer Features

Visualization Geographic information information is often presented as maps. Unfortunately, the browsers can't render maps directly. As a result, often the MapServer combines all geographic layers (vector and image) into a single image (e. g., GIF, JPEG, PNG, SVG, SWF) which is then transmitted to the client for easy visualization in a regular browser (Firefox, IE). However, dedicated clients may offer much more advanced visualization functionality and local manipulation facilities.

Subset Subset or range query is a common operation where the user defined region of interest (ROI) portion of the dataset is returned (as a map) to the user.

Overlay Another typical query where a composite map is generated by overlying the user selected spatial data layers. A subset operation can be combined with an overlay to generate a composite map of a given ROI.

Query Query includes both spatial and non-spatial attributes. Typical spatial queries are point and range queries. In a point query, a spatial object is selected based on the user selected point location (or within a small rectangle about that point); on the other hand, a range query returns all spatial objects within the ROI. Users can also interactively select spatial elements and query the non-spatial attributes associated with the selected spatial objects.

Historical Background

Though there is great public need for geographic information, until recent years, access to digital geographic information has been confined to specialized users. Public access was confined mostly to hard copy products such as paper maps. Initial WebGIS development efforts were concentrated on map visualization and browsing [8]. Query capabilities [7,12] were added soon there after. The earliest WebGISes started to appear in 1995 with the first noticeable contribution being GRASSLinks [9], based on the public domain GIS and image processing system GRASS. The initial response of the geospatial technology industry to the growing popularity of the Internet was to develop Common Gateway Interface (CGI) wrappers to their standalone GIS software. This resulted in thin-client/fat-server systems. In this design, the server is overburdened with both data access and geospatial analysis tasks. The client (typically a Web browser) receives results as HTML pages with embedded image maps. Ignoring performance issues, these systems are a good choice since the user does not need to install any additional resources on his local machine. However, this solution becomes impractical as usage increases, as the server cannot handle a large volume of online users. In recognition of this limitation and with advancements in development environments, the focus of development gradually shifted towards client-side GIS. Client-side GIS is now practical due to various enabling technologies like Java applets, ActiveX Controls, and extensible Web clients. Several researchers have explored client-side GIS architectures and more details can be found in [1]. This approach is very promising and appealing due to advances in Internet programming environments, namely Java. Java provides an architecture-neutral platform which is essential given the heterogeneous hardware and software environment of the Internet. Java applets are a kind of mini application designed to be run in any modern Web browser or applet viewer. Applets are written only once and reside on a server. They will be downloaded, along with the data, upon a user request and are executed on the user machine connected to the Internet with a Java-enabled Web browser.

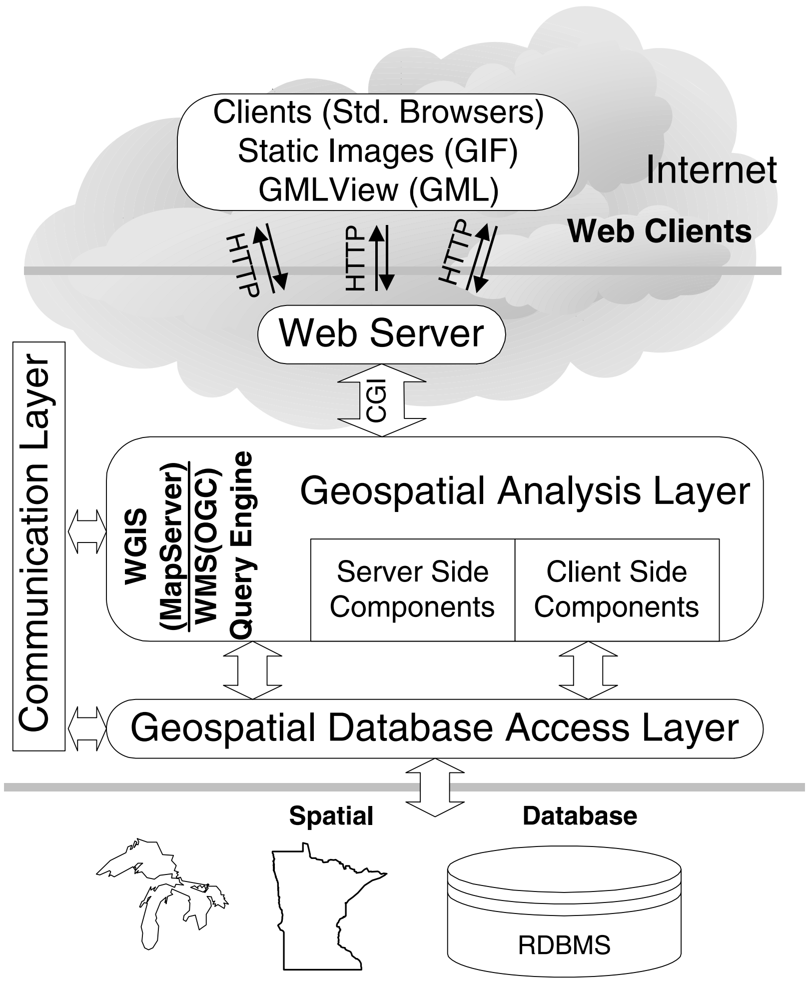
Upon recognizing the need for simple mechanisms to distribute geographic information, the University of Minnesota joined hands with NASA in the early 1990's to develop an efficient means of delivering spatial data products over the Web (ForNet Project [4]). The success of the ForNet project led to a second collaborative research project called TerraSIP [11] between the University of Minnesota, NASA, and a consortium of land management organizations. The tools developed during the ForNet project were eventually culminated into the MapServer. One of the main objectives of TerraSIP was to investigate ways of efficiently implementing a general purpose Internet-based system for the analysis of geospatial data. Several new extensions were developed under this effort, including a load-balancing client-server architecture, an online geo-spatial processing (OLGP [13]), and OGC extensions. The MapServer software is now maintained by a growing number of developers from around the world and is supported by a diverse group of organizations that fund enhancements and maintenance.

Scientific Fundamentals

This section describes the basic building blocks of the MapServer and its support of OGC standards. First, an overview of the application development environment is presented.

MapServer Architecture

MapServer follows a standard 3-tier architecture, as shown in Fig. 1. The three layers are the client, the server, and the geo-spatial database access system (GSDAS). The server layer is further subdivided into the CGI, geo-spatial analysis and communication subsystems. In addition, the architecture of extended MapServer is based on a "balanced client/server" paradigm, as opposed to a client-centric or server-centric approach. First, let us start with the basic MapServer architecture.



University of Minnesota (UMN) Map Server, Figure 1 WMS Compliant MapServer Architecture

Tier 1: The Client In general, the client is any standard Web browser. The application front-end consists of several HTML documents which provide a description of the graphical environment for user interaction with the server. These documents consist of standard HTML tags and JavaScript elements. The browser constructs a Universal Resource Locator (URL) from user input, opens an HTTP connection with the server, and renders the results retrieved from the server. In case of extended MapServer, the client also consists of several applets which not only perform the rendering of geographic elements, but also execute certain geospatial analysis functions. Geospatial analysis is divided between the client and server (Tier 2) based on performance criteria, which makes extended MapServer a well-balanced client/server system.

Tier 2: The Server The server consists of three major components: Web server, application server and MapServer. Web server encapsulates HTTP protocol and services the web client requests. Application server acts as a glue (or middleware) between Web server and other server-side applications such as MapServer. MapServer itself is implemented as a layered architecture.

Layer 1: The CGI Module The CGI module is the component of the application server that responds to HTTP requests submitted by clients and decodes the URL into tokens (i. e., CGI variables). This module is an integral part of MapServer [7]. The parsed tokens are first analyzed by MapServer, then, if any geoprocessing is needed, a request is passed to the geospatial analysis module. The results retrieved from the geospatial analysis module are provided to the client in a form understood by the Web browser.

Layer 2: The GeoSpatial Analysis System Core geographic analysis functionality is implemented in this layer. This module processes the output from the CGI module and sends the request for input data to the communication layer. Although the MapServer is a stand-alone system, it supports distributed spatial databases. The data may not necessarily reside on the same server as the Application Server. Furthermore, the data may not even reside on any single server, but instead may be distributed across several servers.

Layer 3: The Communication System The communication system is responsible for talking to the GeoSpatial Database Servers available in the communication group. It identifies and accesses the required datasets through the GeoSpatial Database System and returns the results to the GeoSpatial Analysis System.

Tier 3: The GeoSpatial Database Access System GSDAS is based on an open architecture which is not tied down to a particular RDBMS or file format. Initial versions of MapServer implemented several native drivers for well-known spatial data formats and RDBMS, however, recent versions of MapServer are integrated with more versatile Open source geospatial data access libraries, such as GDAL and OGR [5].

Client/Server Interaction – A Load-balancing Approach

The distinction between the "thin-client/fat-server" and "thick-client/thin-server" paradigms lies in whether the geo-spatial analysis functionality is supported on the server or the client, respectively. One of the major reasons for the initial "thin-client/fat-server" systems is the fact that HTTP/1.1 is a stateless and connectionless protocol. By connectionless it means that the client and server are connected only for the duration of the transaction. Stateless refers to the fact that the server forgets about the client once the transaction is over. Another reason for the development of these systems is the fact that the Web client does not understand anything about geographic elements. As a result, all of the data access and processing is performed on the server with the results being converted into an image (e. g., JPEG or GIF) or XML and passed to the Web client.

The advancements in development environments have made it possible to extend the Web client in the form of plug-ins and applets. A couple examples of GIS plug-ins are MapGuide [6] from Autodesk and Vdraft [15] from SoftSource. Applets are another popular way of developing client side applications. Plug-ins have the advantage that they can access local data, whereas applets have restricted access to resources on both the client and the server for security reasons. Most advanced clients are now being authored using web 2.0 technologies: DHTML, AJAX, XML, Javascript, and SVG. The main point is that these approaches enable the design of "thin-server/thick-client" systems, facilitating geo-spatial analysis on the client. Under this design, a thin-server can be implemented using some simple CGI scripts which parse the URL to reformulate the client request into a database query or a function call. These calls can then be submitted to a standard GIS or a specifically designed GIS server. Alternatively, one could design a customized GIS server which internally implements HTTP and parses the URL to service the client request (e. g., ArcIMS). These client/server configurations are two extreme approaches to realizing WebGIS. Both place a heavy burden on either the client or the server.

Performance is the most important feature in modern client/server systems [2] and is critical for two reasons: the bandwidth bottleneck of the Internet and the large volume of data to be transmitted in GIS and multimedia applications. Careful study reveals that certain operations are better performed on the server rather than the client and vice versa. This observation led to a new extension of the MapServer which tries to minimize the communication between client and server by implementing the geospatial analysis operation on the server if the resultant output size is less than the input size. In order to load balance the client/server system and to reduce the communication cost, the following general rules have been followed based on observations of NRAMS [13] system performance and analysis of associated MapServer logs:

"Apply geo-spatial analysis function (f_g) on server, (i) if the computational cost (t_f) is less than communication cost (t_c), (ii) if the resultant data (d_o) is less than input data (d_i), otherwise apply the f_g on client."

Based on the status of a run-time parameter defined in the configuration file (named <*projectname*>*.map*), MapServer determines where the given geo-spatial analysis task should be launched. If f_g is computed on the server, then the result is sent back to the client. If f_g is determined to be executed at the client, then the data and the code (applet) are pushed-down to the client. The performance of WebGISes can also be improved using several approaches such as progressive vector transmission, tile and multi-resolution indexing, fast CGI, and automatic load-balancers.

Mapserver Application Basics

This section describes the basic building blocks of any MapServer application. Typically, client interaction with the application front-end (GUI) translates into an URL which gets submitted to the web-server (e. g., Apache). The web-server transfers the request to the application sever (in this case MapServer). MapServer parses the user request (cgi-module) and takes several actions (query processing, data access, output generation, etc.). Appropriate result will then be constructed and returned to the client (user). In oder to understand the user request, MapServer uses a configuration file called .map file. This configuration file, called .map file, is the basic building block of any application.

Application Configuration

MapServer configuration files are used to control virtually all aspects of an application, such as what layers are to be drawn, how they are rendered, and how they are queried. This file also becomes the means for a MapServer application developer to add OGC (WMT, GML, WCS, WFS) compatibility to an application. A MapServer application is configured using several objects, like a Map Object, a Label Object, a Layer Object, a Feature Object, and a Web Object. Each object controls a certain aspect of the application. More detailed descriptions of these objects can be found in the MapServer [7] documentation. Efforts are underway to describe the configuration and data layer definitions using XML. Geo-spatial analysis can be customized by using a separate process configuration file which defines the relationship between a CGI variable and the analysis function to be applied as well as where to do the processing (i. e., on the client or the server). The front-end of the application consists of the graphical user interface, described by standard HTML tags and style sheets. JavaScript is often used for preprocessing complex (dynamic) user requests (e. g., ROI, layers selected, analytical functions), creating new windows and changing the output targets, etc. The CGI module relies on this configuration file to decipher the user request and generate a query execution plan.

MapServer and OGC Standards

Interoperability Recent efforts by the Open GIS consortium have resulted in three different, but related standards that address interoperability, viz. Web Mapping Sever (WMS) [3] (also called as Web Mapping Testbed (WMT)), Geographic Markup Language (GML) [3] and the Web Coverage Service (WCS) [3]. A WMS specification standardizes the way in which maps are requested by clients and the way that servers describe their data holdings. WMS maps are generally rendered in a standard format like Graphics Interchange Format (GIF), Portable Network Graphics (PNG), JPEG, SVG, etc. However, the map based output generated by WMS is often of limited use for other GIS software, apart from serving as a background or reference map layer. On the other hand, GML is an XML encoding for the transport and storage of geographic information, including both the spatial and nonspatial properties of geographic features. The GML specification is more suitable for vector data exchange between WebGISes. More recent efforts to produce a parallel specification for delivering GIS raw data over the Web in a standard way has resulted in the Web Coverage Service (WCS) specification. Owing to its obvious importance in earth sciences data sharing over the Web, the University of Minnesota, along with UND and JPL partners, have developed the WCS extension to MapServer.

WMS Compliant MapServer The original MapServer [7] was designed before OGC specifications were published. The goal of the MapServer/WMS integration effort was to develop native WMS compatibility into the MapServer in such away that would allow users of the CGI application and those writing custom applications using MapScript to take advantage of it. WMT capabilities requests require the presentation of service and metadata. MapServer was designed originally as a map and spatial query engine so generic metadata storage was added to MapServer configuration files to fully support WMT requirements:

```
MAP
  NAME 'example'
  SIZE 300 300
  IMAGECOLOR 255 255 255
  IMAGETYPE PNG
  EXTENT 427400 5228000 451800 5254000
  PROJECTION
    +init=epsg:26915
  END
  WEB
    METADATA
      wms_title 'Example WMS application'
      wms_onlineResource 'http://... '
      wms_contactPerson 'John Doe'
      wms_contactOrganization 'Acme, Inc.'
      ...
    END
  END
  LAYER
    NAME 'water'
    DATA '/usr/data/shapefiles/water'
    CLASS
      COLOR 0 0 255
    END
  END
END
```

Sample MAP file <*projectname.map*>

The WMS functions read the .map file to produce content matching the users WMS request. Any standard browser (or client system) can query a WMS compliant server for one or more of the following services: map images (GetMap), service-level metadata (GetCapabilities), and, optionally, information about particular features (Get-

FeatureInfo). These requests are submitted to the server in the form of URLs. URLs are formed using a set of standard parameters [3](e. g., width, height, bbox, srs, etc.), based on the user interaction with the client system. For example, a request to a AVHRR hurricane image may translate into the following URL for MapServer: http://terrasip.gis.umn.edu/mapserver.cgi?VERSION=1.1.0&REQUEST=GetMap&SRS=EPSG%3A4326&BBOX=-97.105,24.913,78.794,36.358&WIDTH=560&HEIGHT=350&LAYERS=AVHRR-09-27&STYLES=default&FORMAT=image/gif. The resulting spatial data is converted into GIF format (as requested, see FORMAT tag in the above URL) and returned to the client. This representation has the advantage that the maps can be viewed in a standard browser, though client side query processing cannot be achieved.

XML/GML and MapServer XML, the *eXtensible Markup Language*, is fast becoming the standard for data exchange between web applications. HTML (Hypertext Markup Language) and XML are both subsets of SGML (Standard Generalized Markup Language). However, there is a subtle difference between HTML and XML: HTML tags serve the purpose of describing how the data items should be displayed, while XML tags describe the data itself. This difference is very important; the self-describing nature of XML allows the programs to interpret the data in XML documents. Interpretation of data stored in XML documents is possible due to the existence of Document Type Descriptors (DTDs) or XML Schema. GML will have a large impact on the ability of organizations to share geographic information with one another and to enable linked geographic datasets.

GML Processing and Visualization: The basic ideas underlying XML are very simple: tags on data elements identify the meaning of the data rather than, for example, specifying how the data should be formatted (as in HTML) and relationships between data elements are provided via simple nesting and references. Yet, the potential impact is significant: Web servers and applications encoding their data in XML can quickly make their information available in a simple and usable format, one that provides for easy interoperability. Information content is separated from information rendering, making it easy to provide multiple views of the same data. XML is simple, easily parsed, and self-describing. GML is an XML encoding for the transport and storage of geographic information which includes both the geometry and properties of geographic features [3]. The initial release of GML conforms to OGC's "Simple Features" specification. GML provides support for the geometry elements to the corresponding Point, Linestring, LinearRing, Polygon, MultiPoint, MultiLineString, MultiPolygon, and GeometryCollections. It also provides a coordinate element for encoding coordinates and a box element for defining spatial extents. A sample GML fragment for the City layer with simplified schema City(Name: varchar(35), Country: varchar(35), Pop: integer, Shape: Point) is shown below.

```
<city>
    <xsd:Name>Havana</xsd:Name>
    <xsd:Country>Cuba</xsd:Country>
    <xsd:Pop>2</xsd:Pop>
    <gml:Shape>
        <gml:Point gid="P1" srsName="http://
        www.opengis.net/gml/srs/epsg.xml#4326">
            <gml:coord><gml:X>101.8</gml:X>
            <gml:Y>103.4</gml:Y></gml:coord>
        </gml:Point>
        </gml:Shape>
</city>
```

For a given spatial query, MapServer can output a valid GML document. This GML document can then be parsed at the client side to extract the required information. GML Parsing is the process of assigning structure to sentences with a given grammar. In order to demonstrate the usefulness of GML, a Java-based GMLViewer was developed for parsing GML documents returned by the MapServer [10]. This viewer can render vector maps from GML documents on a client machine.

WCS Compliant MapServer Even before the Web Coverage Service (WCS) specification was published, the MapServer design allowed the building of applications that provide web access to raw geo-spatial data products. MapServer has the capability to seamlessly access various data products (of different formats) residing on any server and generate downloadable data products based on the user defined region of interest. However, the generated product was limited to generic binary formats (with BIL, BIP, and BSQ interleaving), with a few well known header formats which allowed users to integrate the downloaded data product into well known commercial and public domain remote sensing and GIS software. Although the data format issues can be addressed with specialized libraries like GDAL, the problem still remains with client-side software systems. First of all, there is no standard way to request data products and secondly, there is no standard way to describe the associated ancillary information (metadata, projection system, etc.). These limitations have hindered MapServer usage as a potential tool to access raw data holdings on distributed servers. This is exactly where the WCS standard comes into play. In a nutshell, MapServer has many components and desirable features that are needed for building a WCS server.

A WCS enabled server provides access to potentially detailed and rich sets of geospatial information. As opposed to the WMS server, which filters and generates data products in the form of static maps, the WCS server actually generates data products that meet user defined criteria in specific (well-known) formats that can be consumed by the client-side software (e. g., ARC/INFO, Imagine, Matlab, IDL). The WCS specification provides the following three operations: GetCapabilities, GetCoverage, and DescribeCoverage.

The *GetCapabilities* operation returns an XML document describing the service and brief descriptions of the data collections from which clients may request coverages. Clients would generally run the GetCapabilities operation and cache its result for use throughout a session or reuse it for multiple sessions. If GetCapabilities cannot return descriptions of its available data, that information must be available from a separate source, such as an image catalog.

The *DescribeCoverage* operation lets clients request a full description of one or more coverages served by a particular WCS server. The server responds with an XML document that fully describes the identified coverages.

The *GetCoverage* operation of a Web Coverage Service is normally run after GetCapabilities and DescribeCoverage replies have shown what requests are allowed and what data is available. The GetCoverage operation returns a coverage (that is, values or properties of a set of geographic locations), bundled in a well-known coverage format. Its syntax and semantics bear some resemblance to the WMS GetMap and WFS GetFeature requests, but several extensions support the retrieval of coverages other than static maps or discrete features.

The GetCapabilites and DescribeCoverage services require generating a good amount of metadata that describes data holdings on the server as well as individual coverages. Metadata is harvested through two distinct sources: i) from the data source itself, either through the header files or through the associated ancillary files (e. g., world file) ii) from a new vocabulary (key-words) which has been added to the MapServer configuration file (.map). Few important WCS modules are described below.

WCS-CGI Module: This module processes the client request with the help of a configuration file with an extension of .map. The key-value pair encoded URL is first scanned by this module. If the request is a valid WCS request, MapServer responds appropriately, otherwise it generates an error message. The configuration file is parsed, and structures are built in memory data which then become key resources for various other modules.

Metadata Harvesting Module: This module is responsible for servicing the GetCapabilites and the DescribeCoverage requests. The output of this module is an XML document that conforms to the WCS 1.0 XML schemas defined in the WCS specification.

Spatio-temporal Tile Indexes: MapServer has long supported a method of breaking a dataset into smaller, more manageable pieces or tiles. A shapefile is used to store the boundary of each tile and an attribute holds the location of the actual data. Within a MapServer mapfile, the layer keywords TILEINDEX and TILEITEM are used to activate tiling. In order to more efficiently support the WCS specification, a new tiling scheme has been implemented within MapServer that supports spatial sub-setting, but also ad hoc sub-setting based on any attributes found within the tile index. The new scheme introduces the concept of tile index layers, that is, a separate layer definition is used to describe the tile index dataset. The syntax for the layer using the index remains unchanged except that the value for TILEINDEX refers to the index layer by name, instead of an external shapefile. More information on WCS extension can be found in [14].

Key Applications

UNEP Geo Data Portal This section describes a few sample applications developed using the MapServer. These descriptions were submitted to the MapServer application gallery by the actual application developers.

UNEP Geo Data Portal GEO Data Portal: This portal (http://geodata.grid.unep.ch/) is the authoritative source for datasets used by UNEP and its partners in the Global Environment Outlook (GEO) report and other integrated environment assessments. Its online database holds more than 400 different variables, as national, subregional, regional and global statistics or as geospatial datasets (maps), covering themes like Freshwater, Population, Forests, Emissions, Climate, Disasters, Health and GDP. One can display them on-the-fly as maps, graphs, raw values or download the data in different formats.

The Atlas of Canada – Major Forest Fires: There are about 9000 forest fires recorded annually in Canada. An average of 2.1 million hectares are burned every year; virtually all of it is boreal forest. Lightning accounts for about 85% of all hectares burned annually; people cause the rest. The fires caused by people are more numerous, but burn a smaller area than those ignited by lightning. The major forest fires map application was built using the MapServer. Through this application, users can interactively browse through and learn more about the major forest fires that have occurred since the early 19th century.

Minnesota DNR Data Deli (http://deli.dnr.state.mn.us/): This award winning application demonstrates many capabilities of the MapServer. The Minnesota Department of Natural Resources GIS Data Deli is an Internet-based spatial data acquisition site that allows users to download raw computer-readable data for use in their GIS, image processing system, or traditional database environment. The site includes links to extensive and summary level data descriptions (metadata) to support users.

Fishing Maps by Angling Technologies (http://angling-technologies.com/): This application is providing access to interactive fishing maps that allow fishermen to visualize lake depths, structure, feeder streams, and any other relevant information. The interactive maps provide users with the capability to design and print their own hard copy maps along with GPS way points which helps them to choose fishing spots. This application demonstrates a wide range of functionality provided through integration of various open source GIS technologies including MapServer, PHP/Mapscript, Chameleon, and PostGIS. Data layers include an extensive integrated collection of Open Source data of benefit to fishermen, including seamless vector layers of TIGER and NHD Water features, DOQs and DRG images.

SEA-COOS Ocean Observations (http://nautilus.baruch.sc.edu/portal_rs/): This application is a portal to near real-time ocean observations, remotely sensed imagery, and ocean model forecasting maps and data. The integration of PHP MapScript, MapServer-CGI, PHP, Perl, NetCDF, PostgreSQL, PostGIS, JavaScript, and external image libraries allows for ad-hoc map generation as well as ad-hoc animations.

Live Weather (http://ows.bom.gov.au/mapserver): This application demonstrates the use of Web Mapping and WMS/WFS functionality for sharing weather data from the Australian Bureau of Meteorology, which traditionally had been distributed in the form of non-referenced images and text.

Antarctic MapViewer (http://www.kgis.scar.org/mapviewer/kgis.phtml): An interactive MapViewer for the Scientific Committee on Antarctic Research's King George Island GIS project (SCAR KGIS). KGIS provides a spatial database for one of the most densely populated areas in Antarctica. The KGIS MapViewer is multi-lingual and allows for customized high quality PDF map output. The KGIS MapViewer is based on php-MapScript and JavaScript.

The MODIS Download Facility (http://terrasip.gis.umn.edu/projects/download/): This application was designed to interactively download MODIS satellite imagery and complimentary vector data in standard formats. Typical MapServers generate only static images which are suitable for visualization. However, by using MapServer, raw data can be distributed in an interactive manner which allows users to select datasets meeting their application requirements (e. g., cloud content and its spatial distribution).

Natural Resource Analysis and Mapping System (NRAMS): The NRAMS application demonstrates the capabilities of providing both online access and analysis to a large collection of multi-temporal AVHRR NDVI imagery along with related spatial and non-spatial datasets for use by land managers and the general public. The analysis techniques developed and incorporated into MapServer allow users to identify vegetation changes over time and across a region, and estimate quantitative biophysical parameters which can be incorporated into global climate models. The NRAMS application can generate on-the-fly temporal profiles for user defined ROIs. A temporal profile is a graphical plot of sequential NDVI observations against time. These profiles quantify the remotely sensed vegetation's seasonality and dynamics. These profiles can be described with simple parameters such as amplitude, mean, and standard deviation. One can understand the onset and peak of greenness and the length of the growing season from analyzing these profiles. By using MapServer, a temporal profile can be generated for a sample location, a window (i. e., an arbitrary rectangular region), or a polygon (e. g., using a county vector layer).

Future Directions

MapServer has always been on the forefront of WebGIS technology, offering features that are not available even in many commercial solutions. The software is constantly evolving with incorporation of new features and standards. The development of MapServer is driven by highly motivated Open source software developers and guided by highly demanding users.

Recommended Reading

1. Abel, D.J., Taylor, K., Ackland, R., Hungerford, S.: An exploration of gis architectures for internet environments. Comp Env Urban Sys **22**(1), 7–23 (1998)
2. Berghel, H.: The client's side of the world wide web. Communications of the ACM **39**(1), 30–40 (1996)
3. Open Geospatial Consortium: Opengis specifications (standards). http://www.opengeospatial.org/standards
4. ForNet: Internet delivery of spatial and related data. http://www.gis.umn.edu/
5. Frank, W.: Gdal – geospatial data abstraction library. http://www.remotesensing.org/gdal/
6. MapGuide. Davecentral: Mapguide plug-in. http://www3.autodesk.com/
7. MapServer. Mapsever home
8. MapViewer. Xerox parc map viewer. http://mapweb.parc.xerox.com/map/
9. REGIS. Grasslinks 3.1. http://www.regis.berkeley.edu/grasslinks/
10. Shekhar, S., Vatsavai, R.R., Sahay, N., Burk, T.E., Lime, S.: Wms and gml based interoperable web mapping system. In ACM-GIS, pp. 106–111 (2001)

11. TerraSIP. Terrasip introduction. http://terrasip.gis.umn.edu/
12. USCB. Tiger mapping service. http://tiger.census.gov/
13. Vatsavai, R.R., Burk, T.E., Wilson, B.T., Shekhar, S.: A web-based browsing and spatial analysis system for regional natural resource analysis and mapping. In: Proceedings of the eighth ACMGIS, Washington D.C., USA (2000)
14. Vatsavai, R.R., Shekhar, S., Burk, T.E., Lime, S.: Umn-mapserver: An high-performance, interoperable, and open source web mapping and geo-spatial analysis system. In: GIScience (2006)
15. Vdraft. Vdraft intenet tools. http://www.vdraft.com/vtools.html

User-Centered

► Geospatial Semantic Web: Personalisation

User Interfaces and Adaptive Maps

L. Tiina Sarjakoski, Tapani Sarjakoski
Department of Geoinformatics and Cartography,
Finnish Geodetic Institute, Masala, Finland

Synonyms

Personalization; Adaptation, complete; Adaptive; Context-aware

Definition

A map is called adaptive if it is able to change its own characteristics automatically according to the user's needs and interests. The user's needs and interests are typically described by context parameters. Some of these parameters are part of the user-profile (language, age, level of skills, etc.), some can be derived automatically (time of usage, location, the device used for interaction, etc.) while the others depend on the task for which the user needs the map (hiking, cycling, etc.). In a web based environment a map is assumed to be displayed to a user by a client application, which receives the pertinent information from a map server as a result of request-response dialog. A web service as such can be a result of combining or chaining multiple web services. For adaptive maps it is necessary to consider in particular three components involved in the service, namely 1) the client application, 2) value-added service, and 3) integrating map-service. The value-added service generates thematic data (layer) to be integrated with or superimposed on top of the background map. The integrating map-service, on the other hand, must be capable of accessing the reference map database or server and integrating the thematic data with the reference map.

Historical Background

At the beginning of the 1990s the research on user interfaces (UIs) focused a great deal on adaptive systems. The work towards adaptive user interfaces has grown from the need to create greater flexibility and user-friendliness in man-machine communication once computers have come to dominate the working environment of many people [8]. The first descriptions and definitions of adaptation came from biology, where adaptation designated any process whereby a structure was progressively modified to give better performance in its environment, see references given by Fairbairn and Erharuyi in [4]. The term 'adaptive map' has earlier been used in such contexts as computational mathematics or computer science. Hypermedia adaptation involved map adaptation technology that comprised various ways of adapting the form of global or local hypermedia maps presented to the user. However, in this entry the focus is on adaptive maps aiming to visualize geospatial information that is delivered from a cartographic web-based service in real-time for mobile or web users.

The rapid development of mobile map applications began in the mid-1990s. The early versions of Location Based Services (LBSs) contained only static images in raster form, and they were adopted from existing digital maps that were originally intended for larger displays. Developers of LBSs, being clearly service oriented, have attached great importance to the end-users. Adaptation of the mobile map applications was seen as a way of improving the usability, see Sarjakoski and Nivala in [6]. Several mobile applications or prototypes that were capable of identifying the location of the user and adapting the presentation of the maps according to the characteristics of the user's mobile device and interests were developed.

One of the major impacts enhancing the development and research on mobile applications with adaptive and context-awareness components was the Information Society Technologies Programme of Research, Technology Development & Demonstration under the 5th Framework Programme of the European Commission. The main focus of the first-year work programme was on enhancing the user-friendliness of the information society: ensuring universally available access and the intuitiveness of next generation interfaces; and encouraging design-for-all. As a result, several mobile applications or prototypes especially for guiding tourists, adaptiveness and context sensitivity as essential ingredients, appeared at the beginning of 21st century.

The introduction of bitmap graphics and the mouse to the personal computer environment by Apple Computer in the early 1980s can be regarded as one of the cornerstones in the progress and evolution of adaptable graphical interfaces [1]. Use of bitmap graphics allows total freedom in the usage of the display area, thus giving an ideal platform support for adaptable user interfaces. The desktop metaphor was introduced at the same time, with the purpose at linking the user experience to those in everyday office life. The usage of desktop metaphor was also supported by guidelines on how to implement the user interfaces in application programs, for giving a consistent user-experience. This already indicated that a good balance should be found between static implementations and adaptation. Bitmap-based touch-sensitive displays have lately arrived at mobile devices, such as the Apple iPhone, making it possible to follow similar principles in the mobile environment as well.

Scientific Fundamentals

Principles for Adaptation

The work towards adaptive user interfaces has grown from the need to create greater flexibility and user-friendliness, but flexibility is also needed for the purposes of system design. It makes the user more independent of the designer, and does not force the designer to decide about user-specific optima before the user works with the system. A flexible system should give the user greater freedom; improving the correspondence between user, task and system characteristics, and increasing the user's efficiency. The goal is to fit the system to a specific user and to a specific task not only on the basis of the requirements analysis made at the design phase, but also by prolonging the flexibility of the system into the usage phase by its adaptation capability. One way at obtaining flexibility is to provide multiple options for functionally equivalent interaction styles [8]. This approach increases the user's freedom, but it also increases the complexity of the system.

Another kind of flexible system is a system that can be tailored. Two variations are known in the literature: adaptable and adaptive systems. A system is called *adaptable* if it provides the end-user with tools that make it possible to change the system characteristics [8]. This kind of individualization gives the user control over the adaptation. It is up to the user to initiate the adaptation and to employ it. There is a wide spectrum of tools and methods in commercial applications for customizing the user interface of the system. For example, special menus or key combinations open up access to interface presentation, naming, and dynamics; there are macro-mechanisms for combining functions.

Several levels of adaptation exist: customization, personalization, or complete adaptation. It is called personalization whenever something is modified in its configuration or behavior by information about the user [14]. A user profile is the central concept in personalization, since in personalization the user makes the system configuration adaptable based on the personal preferences. Personalization and customization are terms often used in marketing, whereas localization is the vocabulary of software engineering [10]. Customization (or tailoring) is used by the service provider to adapt the service for customer needs.

A system is called *adaptive* or self- adapting if it is able to change its own characteristics automatically according to the user's needs. Modification of interface presentation or system behavior depends on the way the user interacts with the system. The system initiates and performs changes appropriate to the user, his tasks, and specific demands. The adaptive system can describe the system characteristics in a system model, the user characteristics in a user model, and the task characteristics in a task model [2,8]. In other words, an adaptive system is based on knowledge about the system, its interface, the task domain, and the user. It must be able to match particular system responses with particular usage profiles.

An essential issue in the adaptation is the question about what is varied, i. e. the adaptation objects. Regarding applications on adaptive maps and user interfaces, the main *adaptation objects* are:

1) User interface including its functionality and design;
2) Geospatial information to be communicated to the user and
3) Visualization of the geospatial information.

Adaptive Map UIs In the best case the user interface of a map application can be adapted to various devices. Compared with indoor usage situations where personal computers or laptops are used, the UI of a mobile device has a limited factor of space, and users must be able to use it while moving. It is also sensitive to light reflections, which must be taken into account in the design. A relatively common UI adaptation is the adaptation to a user language. The language of the interface is personalized according to the given user preferences.

The border between a dynamic map and a UI is increasingly disappearing. Maps have been described as interfaces to geographic information systems (GISs) or interfaces to the real world [9]. Like user interfaces, dynamic maps are also composed of interactive UI elements: links, mouse over legends, pop-up menus, menus in mouses (right button), etc. A map as a UI links the geospatial world to the device or software environment. An example of an integrated environment is a Scalable Vector Graphics (SVG) environment, where graphics in maps and the UI can be integrated.

The functions that are always assumed to exist in map UIs include panning (North-South, East-West) and scaling (zoom-in, zoom-out). A good user interface design principle includes the concept of *consistency*. Consistency is one of the most basic usability principles [7,15]. When users know that the same command or the same function will always have the same effect, they feel more confident in using the system. The same information should be presented in the same location on all screens and dialog boxes, and it should be formatted in the same way to facilitate recognition. UI standards increase the compliance and thereby the consistency. However, the standards do leave the designers with a certain amount of freedom, so standards compliance is not an adequate way to ensure consistency [7]. Consideration of the consistency principle leads to the interesting question of, where to put the limits for adaptation. No doubt, a lot of adaptation capability in the system will always violate the consistency of the user interface.

Adaptation of Geospatial Information In particular, new user groups introduced for mobile map applications, may not necessarily have much previous knowledge of using and interpreting maps. The geospatial information must, therefore, often be strongly generalized and adapted to the specific usage situation, so that information overload can be avoided.

As far as geospatial information is concerned, adaptation can be done separately for value-added information and background (topographic) map information. The most fundamental issue is to delimit the information so that a user will not experience cognitive information overload. Thus, very often efficient *selection* methods have to be applied for the information to be displayed. Adaptation can be used to present the relevant geospatial information for the current user needs:

1) selection of the topographic feature classes or layers (such as roads, buildings, lakes etc.) to be shown on the map;
2) selection of the points of interest (PoI) data (such as value-added data) to be shown on top of the topographic data;
3) selection of relevant text (such as place names) on the current map;
4) specification of the map generalization functions to be executed for the selected features depending on the Level of Detail (LoD) for the information to be presented;
5) specification of other functions to be executed (such as icon placement).

The selection of topographic feature classes is dependent on the database schema that is supported by the map service. The selection is affected by the usage situation of the user, e. g. a mobile user is by default interested only in updated map information which is relevant to the current task.

Adaptive Visualization of Spatial Information The automatically selected geospatial information to be presented on an adaptive map is given a graphical appearance in the visualization process. The map visualization (including colors and line-widths of the feature symbols, types of icon and text) can also be adapted on the map to satisfy the current user needs and other preferences. A typical problem is that mobile devices are used in outdoor usage situations while moving, and therefore brighter colors are needed in maps displayed on them, Fig. 1. The limited space of a mobile screen also sets demands on the level of map generalization to be executed on the map features; it can also be adaptive and is related to the Level of Detail (LoD) preferred in the presentation.

Different kinds of adaptation methods have been proposed in map visualization: highlighting the important features with the aid of different colors, the outlines of features (such as interesting buildings); enhancing the contrast between important and unimportant features; and varying opacity between what is important and unimportant, or animation [10]. Another form of adapting visualization of geospatial information is the selection of map icon types

U

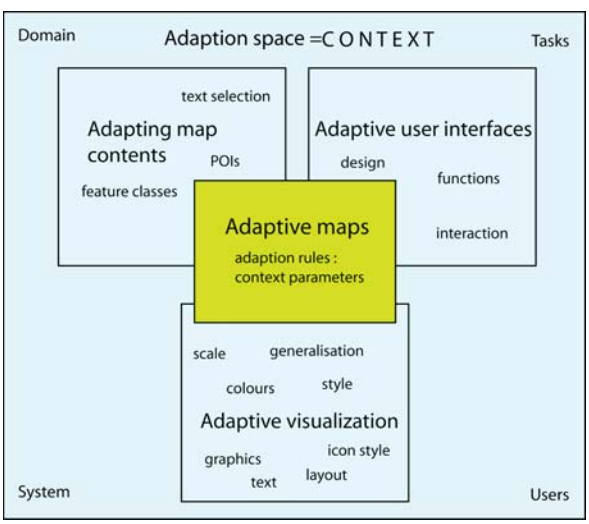
User Interfaces and Adaptive Maps, Figure 1 Adaptation to different devices: a laptop map on the *left* and a PDA map on the *right* according to the user preferences, as shown below in the application examples (Fig. 5).

Context Awareness

Map users desire increasingly intelligent systems that are easy to use, e. g. in car navigation. 'Intelligence' in UIs could be described as a way of making the system aware of the context, adaptive and flexible for different types of users, the usage situation and the devices on which maps are visualized, see Sarjakoski and Nivala in [6]. Adaptiveness and context sensitivity are regarded as essential ingredients for LBSs.

A context has been described as any information that can be used to characterize the situation of an entity, where entity means a person, place or object which is relevant to the interaction between a user and an application, including the user and the applications themselves. A system is described as being context-aware if it uses the context to provide relevant information and/or services to the user, in which the relevancy is dependent on the user's task [3].

More specific context definitions where context is divided into different categories can also be found in literature, see references given by Sarjakoski and Nivala in [6]. For example, the context can be categorized as follows: 1) computing context (such as network connectivity, communication bandwidth and nearby resources such as printers and displays, 2) user context (such as the user's profile, location, people nearby and the current social situation), 3) physical context (lighting, noise levels, traffic conditions and temperature, 4) time context (such as time of day, week, month and season of the year), and 5) context history.



User Interfaces and Adaptive Maps, Figure 2 Relationship between context and adaptive maps

Map adaptation may be controlled by context parameters that form the adaptation space, Fig. 2. Information supplied to the LBSs often relates to the user's immediate vicinity, so the user location is the most prominent context parameter for mobile usage situations. Besides the location, the other contexts relevant to mobile map usage include: orientation of the map, time, navigation history, purpose, social and cultural situation, physical surroundings and system properties.

Key Applications

A number of applications exist where maps and/or their UIs are adapted to different contexts. Some of them are listed here.

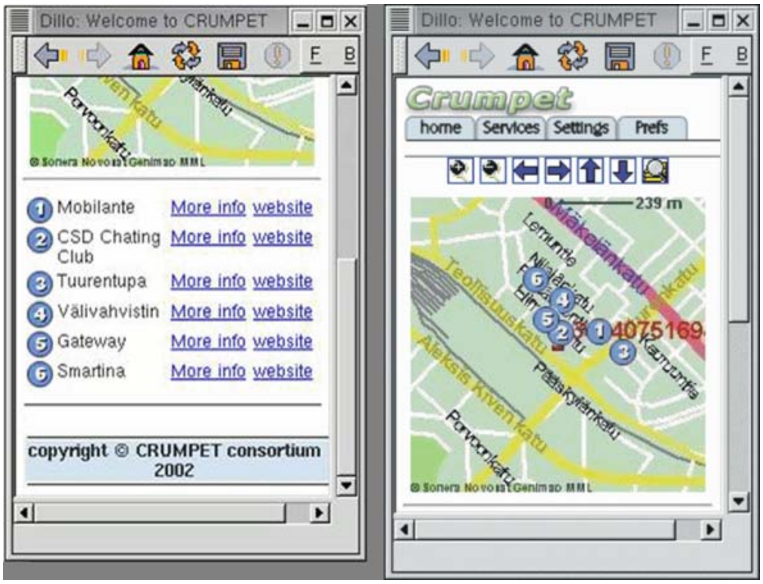
Personalized Services for Tourists

A range of web-based (mobile) geospatial services exist for tourism purposes. In most of the services the location context is used, while other context parameters are more unusual for map adaptation.

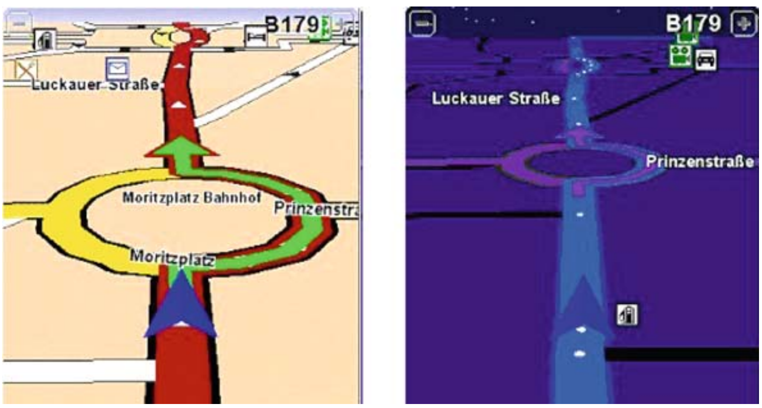
In the GUIDE project the adaptive hypermedia information tailored to the environmental context, as well as the visitor's personal context, was presented to visitors in a tourist office through a browser-based interface, see references given by Baus et al. in [6]. Maps are included in these information spaces, and their choice is, in addition to the other relevant information for the user, adapted by a search on the visitor's approximate position. The mobile component was connected wirelessly to an information server. Based on the current (closest) access point the mobile guide sensed the approximate location. The use of adaptive hypermedia appeared the obvious choice for tailoring the information to the interests of the visitor.

The Deep Map tourist guide was also developed in the LBS context presenting the user location on a mobile map and showing the requested route highlighted and with instructions. Deep Map supported several languages; textual information was stored in multiple languages in a multi-lingual database, see Zipf and Jöst in [4].

In CRUMPET, one of the European Union (EU) funded research projects, a location-aware LBS for tourism including personalized user interaction, was developed, see references given by Baus et al. in [6]. An adaptation of services and information contents to the user's personal interests and their current location was seen as a way of filtering the amount of available information (Fig 3). The filtering process was based on a user profile describing the interests, abilities and characteristics of the user. An agent-based technology was used for development.



User Interfaces and Adaptive Maps, Figure 3 CRUMPET, a location-aware LBS enhancing the requested route and PoIs for tourists, © TeliaSonera



User Interfaces and Adaptive Maps, Figure 4 An egocentric presentation of the TomTomGO navigator showing adapted maps for day/night driving (tomtom.com)

Other lines of research have included the incorporation of multimedia technology. For example, in Navio project multimedia cartography presentations were used to support wayfinding and navigation for pedestrians, see Gartner et al. in [4]. Pedestrian guidance was improved by embedding active and/or passive landmarks. Landmarks are prominent objects that act as marks for places and can be used as reference points. In the ReGeo IST-project a multimedia-based system for tourist guidance (in particular for hikers and skiiers) including 2D and 3D maps and satellite images was developed, see Luley et al. in [4]. The TellMaris research prototype provided 3D mobile map views for boat tourists, see Schilling et al. in [6].

Navigational Applications

Services for different types of navigators for drivers is perhaps the most common application areas. Most of the applications are for car navigation purposes, but there are also navigational guides for cyclists and pedestrians. Examples on navigational applications including locational context awareness are: Navman GPS 3300 Terrain, Outdoor Navigator, TomTom CityMaps, Falk City Guide and MapWay. In the TomTomGO application time context (daytime maps / night time maps) besides the locational component has been used into which maps can be adapted (Fig.4).

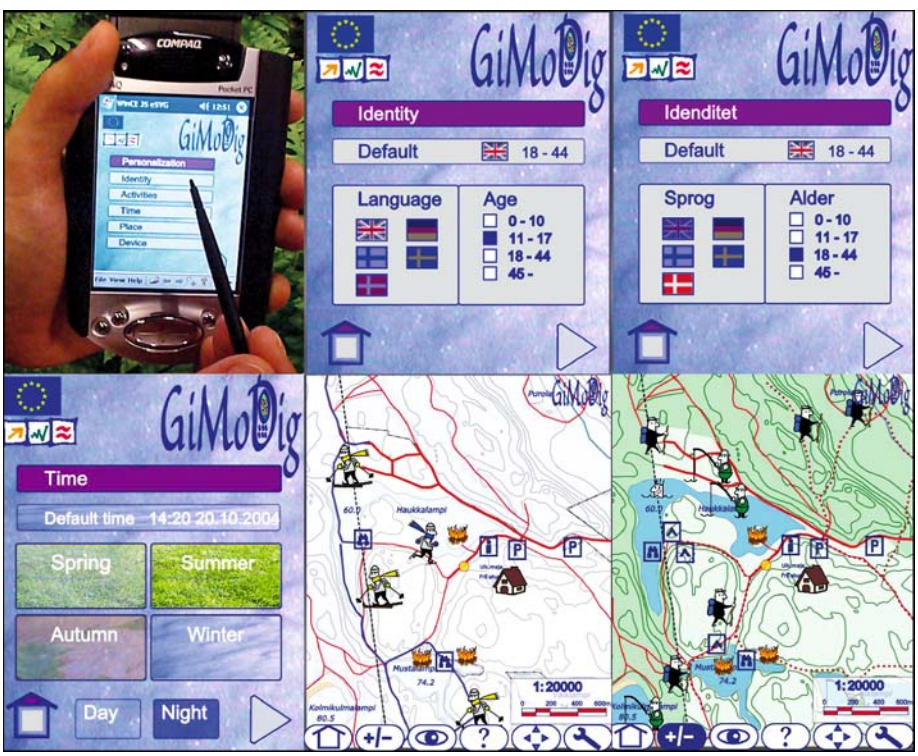
Map Services with Adaptation

One example of projects of the IST program is the GiMoDig project [12]. This project is discussed more deeply in the following in order to give an insight into map services with adaptation capacity. In the project research was carried out on map services for mobile applications, including not only real-time data-integration and generalization but also adaptive maps and user interfaces. A comprehensive research prototype was implemented for verifying the concepts. The following system components are essential in adaptation: 1) an end-user client application, 2) a value-added service and 3) an integrating map service. The end-user application can be a map application on a map-based client on a mobile phone, on a Personal Digital Assistant (PDA) or on a PC in a web-browser environment. This versatility as such created a demand for the service components in the system to be adaptive, i. e. to be able to adapt the map to be suitable for these different devices and usage conditions.

In the system architecture the value-added service is a portal for a client, i. e. a client sends a request to the value-added service and returns a map as a response. For value chains this service is vital. The service has to be adapted to fit the specific needs of the users. In map-based services, information contents are of primary importance. This content can be seen to consist of thematic information and a background map. The selection of thematic information in particular must be based of the user's needs. Typically the thematic information is also very dynamic and updated continuously in the databases, making on-the-fly map creation and display a true necessity. Besides the issues related to information content, the visualization parameters also have to be adapted to serve the user needs.

These aspects – content of the thematic information and background map, and the visualization rules for both of these – are defined in the GiMoDig prototype in the value-added service by its system administrator in such a way that dynamic mapping based on context parameters (describing the user needs) is facilitated. The definitions are implemented in a kind of knowledge base, and the creation or update of this knowledge base is a process in which the system administrator adapts the system to the specific needs of the application [11]. The knowledge base for each distinct application area (such as cycling, hiking) is an *adaptation* process: the system administrator is changing or tailoring the functionality of the system. An integrating map-service is needed in this approach for carrying out the technical process of compiling and rendering the desired map, based on a request from the value-added server. The data for the background map has to be requested from an appropriate source and integrated with the value-added thematic information. This process comprises techniques such as on-the-fly generalization, icon placement, and conversion to graphical form, following the visualization rules in the request [13]. In GiMoDig this process chain was implemented using several layers, while this is an implementation rather than an essential issue for adaptation.

Figure 5 shows that after personalization of the service, the GiMoDig service delivers adaptive maps in real-time to the mobile device. The maps are automatically adapted to the



User Interfaces and Adaptive Maps, Figure 5 The GiMoDig service delivers adaptive maps in real-time to the mobile devices

User Interfaces and Adaptive Maps, Figure 6 A variable-scale map [5]

different users in different usage situations. Here the contents and visualization of seasonal maps (summer/winter) are adapted for teenagers who aim to go hiking/skiing. The figure shows additionally that the user interface can be adapted to five different user languages.

Other Examples

An example of an adaptation of a map is also a so-called variable-scale map [5]. This approach applies adaptive generalization, based on the current location of the user. In a variable scale map for small mobile devices, the level of map generalization increases towards the edges of the map, Fig. 6.

One more example of adaptive maps is an egocentric map, resulting from the increasing personalization capacities of map services. An egocentric map is known as a map depicting geospatial information from an individual user's position (Fig. 4), and the individual user profile is embodied as the ego center in the corresponding map. There is a growing demand for egocentric maps, since in most of the cases the mobile maps are location based and map use is always a personal activity, see Meng in [6].

Future Directions

It is likely that it will become very natural for people to use handheld devices for personal navigation and finding location of interesting places. Consumers will also accept marketing services that alert them to special offers when they arrive in certain places. Permission-based tracking of friends and family will become an accepted part of life and, as most phone calls to the emergency services are made from mobile phones, location information will be routinely and automatically captured, so that people at risk can be found quickly and reliably. As a follow-up to this trend, proper and robust solutions have to be found for human-computer interaction.

Until now the preference has been for adaptation to be mainly a user-intitiated activity in order to achieve controllability and consistency. It is likely that in the future, it will be common to depart from this trend towards acceptance of truly intelligent user interfaces, in which the system will become more proactive in its behavior. These will be supported with such technologies as voice control, tactile and sensor-supported input/output devices, and wearable computing devices. This altogether will promote the use of adaptive maps for LBSs and services embedded into everyday-life environments in various exciting, new ways, taking advantage of the emerging technologies, such as ubiquitous and pervasive computing, ambient informatics and tangible media.

Cross References

► Mobile Usage and Adaptive Visualization

Recommended Reading

1. Apple. Human Interface Guidelines: The Apple Desktop Interface, Apple Computer, Inc. Addison-Wesley Publishing, Reading, MA (1987)
2. Browne, D., Totterdell, P., Norman, M. (eds.): Adaptive User Interfaces. Computers and People Series. Academic Press, London (1990)
3. Dey, A.K.: Understanding and Using Context. Pers. Ubiquitous Comput. **5**(1), 4–7 (2001)
4. Gartner, G. (ed.): Location Based Services & Telecartography, Proceedings of the Symposium 2004, January 28–29, Geowissenschaftliche Mitteilungen. number 66, Technische Universität Wien, Wien (2003)
5. Harrie, L., Sarjakoski, L.T., Lehto, L.: A Mapping Function for Variable-Scale Maps in Small-Display Cartography. J. Geospa. Eng. **4**(2), 111–123 (2002)
6. Meng, L., Zipf, A., Reichenbacher, T. (eds.): Map-based Mobile Services, Theories, Methods and Implementations Springer, Berlin Heidelberg, New York (2005)
7. Nielsen, J.: Usability Engineering. Morgan Kaufmann, New York (1993)
8. Oppermann, R. (ed.): Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software. Lawrence Erlbaum Associates, Hillsdale, NJ (1994)
9. Peterson, M.P.: Interactive and Animated Cartography. Prentice Hall, Englewood Cliffs, New Jersey (1995)
10. Reichenbacher, T.: Mobile Cartography — Adaptive Visualisation of Geographic Information on Mobile Devices. Dissertation. Department of Cartography, Technische Universität München. Verlag Dr. Hut, München (2004)
11. Sarjakoski, L.T., Koivula, T., Sarjakoski, T.: Knowledge-Based Map Adaptation for Mobile Map Services. In: Gartner, G., Cartwright, W., Peterson, M.P. (eds.) Location Based Services and TeleCartography. Lecture Notes in Geoinformation and Cartography. Springer, Berlin Heidelberg New York (2007), pp. 247–264

12. Sarjakoski, T., Sarjakoski, L.T.: The GiMoDig public final report. Geospatial info-mobility service by real-time data-integration and generalisation, GiMoDig-project, IST-2000-30090, Deliverable D1.2.31, Public EC report, 70 pages. http://gimodig.fgi.fi/deliverables.php (30/03/07), 2005
13. Sarjakoski, T., Sarjakoski, L.T.: A Real-Time Generalisation and Map Adaptation Approach for Location-Based Services. In: Mackaness, W.A., Ruas, A., Sarjakoski, L.T. (eds.) Generalisation of Geographic Information: Cartographic Modelling and Applications. International Cartographic Association, Elsevier (2007), pp. 137–159
14. Searby, S.: Personalisation – an overview of its use and potential. BT Tech. J. **21**(1), 13–19 (2003)
15. Shneiderman, B.: Designing the User Interface, Strategies for Effective Human-Computer Interaction, 3rd edn. Addison-Wesley Longman, Inc., Reading, Massachusetts (1998)

VA-File

► Indexing, High Dimensional

Vague Boundaries

► Objects with Broad Boundaries

Vague Spatial Data Types

ALEJANDRO PAULY, MARKUS SCHNEIDER
Department of Computer and Information Science and Engineering (CISE), University of Florida, Gainesville, FL, USA

Synonyms

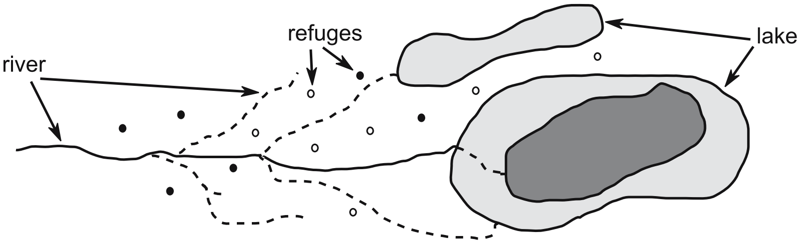
Data types for uncertain, indeterminate, or imprecise spatial objects; Point, vague; Point, kernel; Point, conjecture; Region, conjeccture; Operation, vague spatial

Definition

Naturally occurring phenomena in space often (if not always) cannot be precisely defined because of the intrinsic uncertainty of their features. The location of animal refuges might not be precisely known, and the path of rivers might be uncertain due to water volume fluctuations and changing land characteristics. The extension of lakes can also change and thus have uncertain areas. All these are examples of *vague spatial objects*. The animal refuge locations can be modeled as a *vague point* object where the precisely known locations are called the *kernel point* object and the assumed locations are denoted as the *conjecture point* object. The river paths can be modeled as *vague line* objects. Some segments or parts of the path, called *kernel line* objects, can be definitely identified since they are always part of the river. Other paths can only be assumed, and these are denoted as *conjecture line* objects. Knowledge about the extension of lakes can be modeled similarly with *vague regions* formed by *kernel* and *conjecture* parts. Figure 1 illustrates the examples above. Dark shaded areas, straight lines, and black points indicate kernel parts; areas with light gray interiors, dashed lines, and hollow points refer to conjecture parts.

As another example, consider a homeland security scenario in which secret services (should) have knowledge of the whereabouts of terrorists. For each terrorist, some of their refuges are precisely known, some are not and only conjectures. These locations can be modeled as a *vague point* object where the precisely known locations are represented by the kernel part of the object and the assumed locations are denoted as its conjecture part. Secret services are also interested in the routes a terrorist takes to move from one refuge to another. These routes can be modeled as *vague line* objects. Some routes, represented by the kernel part of the object, have been identified. Other routes can only be assumed to be taken by a terrorist; they are denoted as the conjecture part of the object. Knowledge about areas of terrorist activities is also important for secret services. From some areas it is well known that a terrorist operates in them. These areas are denoted as the kernel parts. From other areas it can only be assumed that they are the target of terrorist activity, and they are denoted as the conjecture parts. Figure 1 gives some examples. Grey shaded areas, straight lines, and gray points indicate kernel parts; areas with white interiors, dashed lines, and white points refer to conjecture parts.

The definition of vague points, vague lines, and vague regions leverages the data types *point* for crisp points, *line* for crisp lines, and *region* for crisp regions. All crisp spatial data types $\alpha \in \{point, line, region\}$ are assumed to have a complex inner structure as it has been defined in [4]. In particular, this means that a *point* object includes a finite number of single points, a *line* object is assembled from a finite number of curves, and a *region* object consists of a finite number of disjoint faces possibly containing a finite number of disjoint holes. Further, these types must be closed under the geometric set operations *union* ($\oplus$: $\alpha \times \alpha \to \alpha$), *intersection* ($\otimes$: $\alpha \times \alpha \to \alpha$), *difference* ($\ominus$: $\alpha \times \alpha \to \alpha$), and *complement* ($\sim$: $\alpha \to \alpha$). Each type



Vague Spatial Data Types, Figure 1 Examples of a (complex) vague point object representing the animal refuges, a (complex) vague line object as a river, and a (complex) vague region object representing a lake

α together with the operations $\oplus$ and $\otimes$ forms a boolean algebra. The identity of $\otimes$ is denoted by **1**, which corresponds to $\mathbb{R}^2$. The identity of $\oplus$ is presented by **0**, which corresponds to the empty spatial object (empty point set).
A vague spatial object is defined by a pair of two *disjoint* or *meeting* crisp complex spatial objects [5]. The extension of a crisp spatial data type to a corresponding vague type is given by a type constructor v as follows:

$$v(\alpha) = \alpha \times \alpha\,, \quad \forall\alpha \in \{point, line, region\}\,.$$

This means that for $\alpha = region$ the type $v(region) = region \times region$, which is also named *vregion* is defined. Accordingly, $v(line) = line \times line$ and $v(point) = point \times point$ define *vline* and *vpoint* respectively. For a vague spatial object $A = (A_k, A_c) \in v(\alpha)$, the first crisp spatial object A_k, called the *kernel part*, describes the determinate component of A, that is, the component that definitely and always belongs to the vague object. The second crisp spatial object A_c, called the *conjecture part*, describes the vague component of A, that is, the component for which it cannot be said with any certainty whether it or subparts of it belong to the vague object or not. *Maybe* the conjecture part or subparts of it belong to the vague object, *maybe* this is not the case. Since the kernel part and the conjecture part of the *same* vague spatial object may not share interior points a restriction is imposed to assure that the interior point sets[1] do not intersect, formally:

$$\forall\alpha \in \{point, line, region\}$$
$$\forall A = (A_k, A_c) \in v(\alpha) \colon A_k^\circ \cap A_c^\circ = \emptyset\,.$$

Hence, A_k can be regarded as a lower (minimal, guaranteed) approximation of A and $(A_k \oplus A_c)$ can be considered as an upper (maximally possible, speculative) approximation of A.

Historical Background

Spatial vagueness has to be seen in contrast to spatial uncertainty resulting from either a lack of knowledge about the position and shape of an object (*positional* uncertainty) or the inability of measuring such an object precisely (*measurement* uncertainty). Much literature has been published on dealing with positional and measurement uncertainty; it mainly proposes probabilistic models. Spatial vagueness is an intrinsic feature of a spatial object for which it cannot be said whether certain components belong to the spatial object or not. The design goal for dealing with spatial vagueness in VASA is to base the definition of vague spatial data types and their operations on already existing definitions of exact spatial objects. This so-called exact model approach is also followed in the definition of broad-boundary regions [1] and the egg-yolk approach [2] as it is detailed in [this same chapter]. A generalization of the ideas from the broad-boundary approach can be found in the original definition of *vague regions* [3]. This definition proposes a data type for vague regions that is closed under the union, intersection, difference, and complement operations. The components of VASA are based on the original vague regions concept which which is generalized in order to deal with vague points and vague lines.

Scientific Fundamentals

One of the major objectives of exact model based design is to make use of the formalisms introduced by the underlying models upon which the design is based. This allows the new design to relay the major responsibilities of robustness and correctness to the underlying model. A side effect of this type of design is the centralization of the mathematical definitions that form the core of both the underlying model and the new model. The result is a modular design that enables more robust and less error prone specifications.
In the next section the proper definitions of vague spatial operations are formalized. Further details, specifically in what relates to topological predicates between vague spatial objects can be found in [6,7].

Vague Spatial Operations

The three vague geometric set operations **union**, **intersection**, and **difference** have all the same signature $v(\alpha) \times v(\alpha) \to v(\alpha)$. In addition, the operation **complement** is defined with the signature $v(\alpha) \to v(\alpha)$. All of these

[1] x° is used to denote the interior point set of crisp spatial object x

operations are defined in a type-independent and thus generic manner. In order to define them for two vague spatial objects u and w, it is helpful to consider meaningful relationships between the kernel part, the conjecture part, and the outside part of u and w. For each operation a table is given where a column/row labeled by k, c, or o denotes the kernel part, conjecture part, or outside part of u/w. Each entry of the table denotes a possible combination, i. e., intersection, of kernel parts, conjecture parts, and outside parts of both objects, and the label in each entry specifies whether the corresponding intersection belongs to the kernel part, conjecture part, or outside part of the operation's result object.

The *union* (Table 1) of a kernel part with any other part is a kernel part since the union of two vague spatial objects asks for membership in either object and since membership is already assured by the given kernel part. Likewise, the union of two conjecture parts or the union of a conjecture part with the outside should be a conjecture part, and only the parts which belong to the outside of both objects contribute to the outside of the union.

The outside of the *intersection* (Table 1) is given by either region's outside because intersection requires membership in both regions. The kernel part of the intersection only contains components which definitely belong to the kernel parts of both objects, and intersections of conjecture parts with each other or with kernel parts make up the conjecture part of the intersection.

Obviously, the *complement* (Table 1) of the kernel part should be the outside, and vice versa. With respect to the conjecture part, anything inside the vague part of an object might or might not belong to the object. Hence, it cannot be said with certainty that the complement of the vague part is the outside. Neither can be said that the complement belongs to the kernel part. Thus, the only reasonable conclusion is to define the complement of the conjecture part to be the conjecture part itself.

Vague Spatial Data Types, Table 1 Components resulting from intersecting kernel parts, conjecture parts, and outside parts of two vague spatial objects with each other for the four vague geometric set operations

union	*k*	*c*	*o*
k	*k*	*k*	*k*
c	*k*	*c*	*c*
o	*k*	*c*	*o*

intersection	*k*	*c*	*o*
k	*k*	*c*	*o*
c	*c*	*c*	*o*
o	*o*	*o*	*o*

difference	*k*	*c*	*o*
k	*o*	*c*	*k*
c	*o*	*c*	*c*
o	*o*	*o*	*o*

complement	*k*	*c*	*o*
	o	*c*	*k*

The definition of *difference* (Table 1) between u and w can be derived from the definition of complement since it is equal to the intersection of u with the complement of v. That is, removing a kernel part means intersection with the outside which always leads to outside, and removing anything from the outside leaves the outside part unaffected. Similarly, removing a conjecture part means intersection with the conjecture part and thus results in a conjecture part for kernel parts and conjecture parts, and removing the outside of w (i. e., nothing) does not affect any part of u.

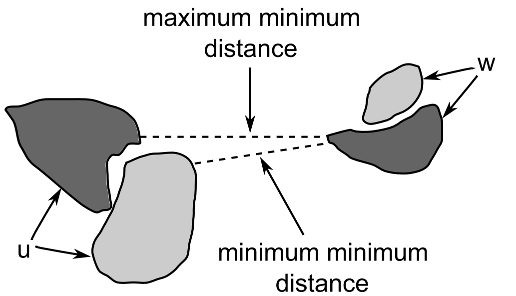
Motivated by the intended semantics for the four operations described above, the formal definitions are provided. An interesting aspect is that these definitions can be based solely on already known crisp geometric set operations on well-understood exact spatial objects. Hence, *executable specifications* can be defined for the vague geometric set operations. This means, once having the implementation of a crisp spatial algebra available, it can directly *execute* the vague geometric set operations without being forced to design and implement new algorithms for them.

Let $u, w \in v(\alpha)$, and let u_k and w_k denote their kernel parts and u_c and w_c their conjecture parts:

$$\begin{aligned} u \textbf{ union } w &:= (u_k \oplus w_k, (u_c \oplus w_c) \ominus (w_k \oplus w_k)) \\ u \textbf{ intersection } w &:= (u_k \otimes w_k, (u_c \otimes w_c) \oplus (u_k \otimes w_c) \\ &\quad \oplus (u_c \otimes w_k)) \\ u \textbf{ difference } w &:= (u_k \otimes (\sim w_k), (u_c \otimes w_c) \\ &\quad \oplus (u_k \otimes w_c) \oplus (u_c \otimes (\sim w_k))) \\ \textbf{complement } u &:= (\sim u_k, u_c) \,. \end{aligned}$$

Spatial operations that result in unique numeric values can be applied to vague spatial objects generally by transforming the result into ranges of values. That is, the operations are specified as executions of their crisp versions returning a lower bound result and an upper bound result. These values depend on whether the conjecture parts are considered in the computation or not.

To compute the minimum distance between two vague spatial objects, define the operations *vague-min-mindistance* $: v(\alpha) \times v(\beta) \to real$ and *vague-max-mindistance* $: v(\alpha) \times v(\beta) \to real$ are defined. Both operations can be applied to pairs of vague spatial objects of possibly distinct types (that is, $(\alpha = \beta \vee \alpha \neq \beta)$ and $\alpha, \beta \in \{point, line, region\}$). The first operation considers all kernel and conjecture parts, thus returning the minimum possible distance between both objects. The second operation, only considers kernel parts, thus returning the maximum possible minimum distance between both objects that is, the maximum value that the minimum distance will actually be. An illustration of the maximum minimum distance and minimum minimum distance between two vague regions is shown in Fig. 2. Formally, let *mindistance* $: \alpha \times \beta \to real$



Vague Spatial Data Types, Figure 2 An example illustrating the maximum and minimum minimum distances between two vague spatial regions. The dark shaded areas conform the kernel parts of the objects and the light shaded areas represent the conjecture parts

be the minimum distance operation defined for crisp spatial objects:

$$\begin{aligned} \textit{vague-min-mindistance}(u, w) &:= \textit{mindistance}((u_k \oplus u_c), (w_k \oplus w_c)) \\ \textit{vague-max-mindistance}(u, w) &:= \textit{mindistance}(u_k, (w_k)) \, . \end{aligned}$$

Unary numeric operations are used to express properties of a vague spatial object. The operations *min-length* : *vline* → *real* and *max-length* : *vline* → *real* are defined to compute the range of the length of a vague line object. The operations *min-area* : *vregion* → *real* and *max-area* : *vregion* → *real* are used to compute the area of a vague region. Inversely to the distance operation, the minimum length (area) of a vague line (region) is computed by taking into consideration all parts, including the conjecture part of the object. The maximum length (area) is computed by only considering the kernel part of the object. Formally, let *length* : *line* → *real* and *area* : *region* → *real* refer to the operations that compute the length and area of a crisp line and region respectively. Also consider $a \in vline$ and $b \in vregion$:

$$\begin{aligned} \textit{min-length}(a) &:= \textit{length}(a_k) \\ \textit{max-length}(a) &:= \textit{length}(a_k \otimes a_c) \\ \textit{min-area}(b) &:= \textit{area}(b_k) \\ \textit{max-area}(b) &:= \textit{area}(b_k \oplus b_c) \, . \end{aligned}$$

The definitions provided above serve as a sample of the operations that can be defined for vague spatial objects as an executable specification of operations on the underlying crisp spatial objects.

Key Applications

Generally, because many GIS applications largely deal with naturally occurring spatial phenomena that often contain implicit uncertain features, they will all benefit from data models that include considerations for dealing with spatial vagueness. Experts from a wide range of domains such as biology and agriculture can begin to take into account the inexact data that can make a difference in their decision making process. The following three applications are just examples of the wide range of domains that can benefit from dealing with vague spatial data.

- **Ecology:** Ecologists require an abundance of data related to the distribution and interactions of living organisms in their environment. The vast majority of these data suffers from indeterminacy stemming not only from its implicit nature but also from the inability to process exact observations. As a result many of the data is inferred or approximated from actual observations and thus must be treated as uncertain.
- **Military:** Military operations are often designed on the basis of intelligence collected on-site or remotely via technological media. It is also often the case that the intelligence is vague because only pieces of information can be collected or because the sources are not trustful (amongst other reasons).
- **Soil Sciences:** Soil variability is often a problem that must be taken into account when treating soil related data. Fine grained modeling of soil data turns out to be very costly, due to its high variability and inhomogeneity. As a result, in many applications within soil studies, it is enough to approximate the composition of soil to vague models in order to generate the necessary information.

Future Directions

As new data models are generated, the main element upon which their popularity depends is the availability of appropriate data sets. VASA provides a data model appropriate for dealing with uncertainty of spatial data. To motivate its future use, it is necessary to collect data in the appropriate format so that it can be fully exploited by the data model.

Recommended Reading

1. Clementini, E., Di Felice, P.: An Algebraic Model for Spatial Objects with Indeterminate Boundaries. In: Geographic Objects with Indeterminate Boundaries, pp. 153–169. Taylor & Francis, London, UK (1996)
2. Cohn, A.G., Gotts, N.M.: The 'Egg-Yolk' Representation of Regions with Indeterminate Boundaries. In: Geographic Objects with Indeterminate Boundaries, pp. 171–187. Taylor & Francis, London, UK (1996)
3. Erwig, M., Schneider, M.: Vague Regions. 5th Int. Symp. on Advances in Spatial Databases, pp. 298–320. Springer-Verlag, Berlin, Heidelberg, Germany (1997)
4. Schneider, M., Behr, T.: Topological Relationships between Complex Spatial Objects. ACM Trans. on Database Sys. (TODS) **31**, 39–81 (2006)

5. Pauly, A., Schneider, M.: Vague Spatial Data Types, Set Operations, and Predicates. 8th East-European Conf. on Advances in Databases and Information Systems. Budapest, Hungary 22–25 Sep 2004
6. Pauly, A., Schneider, M.: Topological Predicates between Vague Spatial Objects. 9th Int. Symp. on Spatial and Temporal Databases. Angra dos Reis, Brazil (2005)
7. Pauly, A. Schneider, M.: Topological Reasoning for Identifying a Complete Set of Topological Predicates between Vague Spatial Objects. 19th Florida Artificial Intelligence Research Society (FLAIRS) Conference. (2006) pp. 731–736. Melbourne Beach, Florida, USA

Vagueness

► Uncertainty, Semantic

Variogram

► Kriging

Variogram Modeling

► Semivariogram Modeling

Vector Data

VIJAY GANDHI
Department of Computer Science and Engineering,
University of Minnesota, Minneapolis, MN, USA

Synonyms

Geometric modeling; Vector graphics

Definition

Spatial data is data related to a location. Some examples include the population of a city, the type of soil in a region, and data from remote-sensing satellites. In the first example, the city could be considered as a location and the population is the data or a feature. In the second example, the region is a collection of locations and the type of soil is the feature. Note that a location may have one or more features. For example, it may be useful to represent both the population and the average age group of a city.

It is necessary to convert spatial data into a form that a computer can understand. Both models of the data must have the property of storing locations, features and the association between the two. There are two major ways to model spatial data: as vector data or as raster data.

Vector data involves storing data as geometric objects. For example, a road can be represented as a combination of lines. In this case, the lines are the objects. A given real-world situation can be represented as either a raster or a vector model.

The choice between a raster data model and vector data depends on the conceptualization of the feature. There are certain properties of vector data that make them a better choice to represent the model. Vector data is more useful for data that can be represented as discrete objects. Also, it is easier to handle on computers and it takes less space.

Here the focus is on the properties of the vector data with examples and a comparison with raster data.

Historical Background

Vector graphics was probably the earliest field to make use of vector data. Vector graphics have been popular since the CRT display monitors back in the 1950s. To create an image on a screen, CRT monitors trace a beam on to a monitor's surface. The path traced by the beam was a line; in mathematical terms, a vector. These monitors produced high-resolution images. Since then, vector data become known as being of good quality.

Scientific Fundamentals

Representation of the Vector Data

Vector data can be thought of as an object described using mathematical notations. Vector data is represented as a collection of simple geometric objects such as points, lines, polygons, arcs, circles, etc. For example, a city may be represented by a point, a road may be represented by a collection of lines, and a state may be represented as a polygon.

Consider the aerial image of a geographical area shown in Fig. 1:

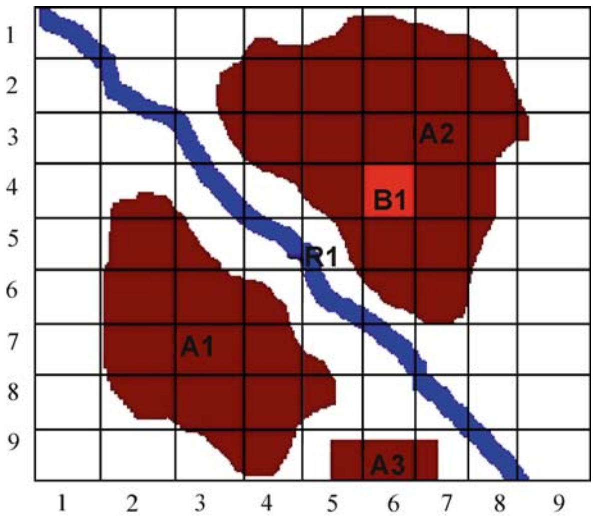
The area shown in Fig. 1 consists of a few entities, namely, a river (R1), a building (B1) and three patches of land (A1, A2, and A3). To represent Fig. 1 in vector data, each entity would be represented by an object. For example, the building B1 could be represented by a point, the river R1 by a line string, and the patch of land A1 by a polygon. The graphical representation of these objects is given in Fig. 2:

Mathematically, these objects could be expressed as:

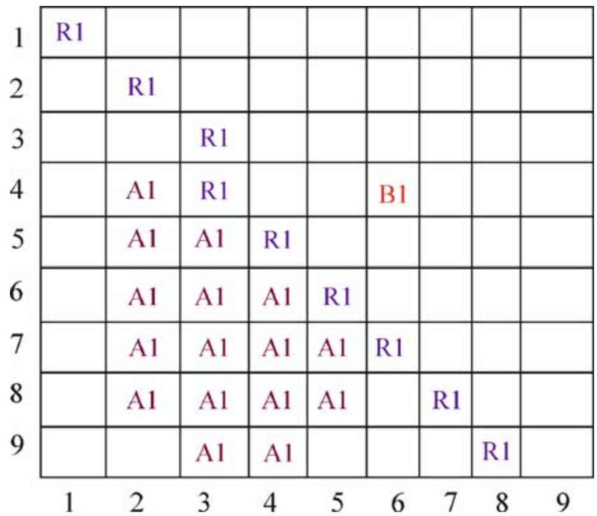
- Building B1: point (6,4)
- River R1: linestring [(1,1), (3,3), (3,4), (8,9)]
- Area A1: polygon [(2,4), (2,8), (3,9), (4,9), (5,8), (5,7)]

In the backend, each object ID and its corresponding points are stored.

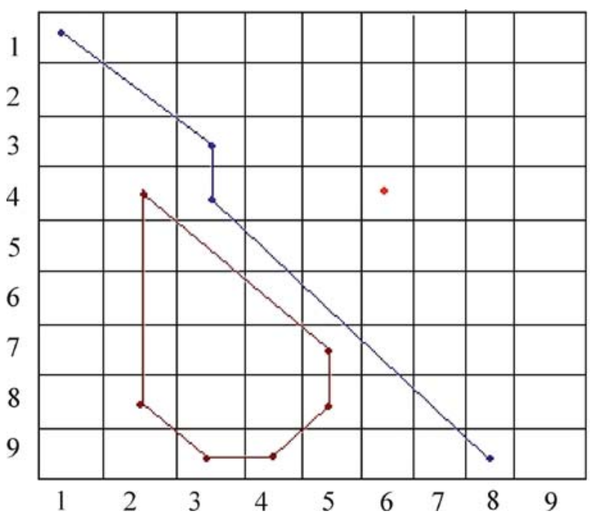
The same map could be represented in a raster data model as shown in the Fig. 3.



Vector Data, Figure 1 Aerial image of a geographical area. *R1* River, *B1* building, *A1*, *A2*, *A3* patches of land



Vector Data, Figure 3 Graphical representation of Raster data model for B1, R1 and A1 shown in Fig. 1



Vector Data, Figure 2 Graphical representation of the vector data for B1, R1 and A1 shown in Fig. 1

The image is divided into a grid and at each pixel the value corresponding to the corresponding object is stored. Hence the pixel (6,4) belongs to object B1; pixel (8,9) belongs to R1, etc.

In the back-end, the pixel-value pair is stored. Hence all the pixels (in this case 9×9) would be stored.

Vector Data and Storage

Since geographical data involves generally involves millions of pixels of data, storage requirement is one of the main considerations in choosing a data model. Generally, the amount of memory required by a vector model would be less than that of a raster model, because the data could be emulated by the vectors. Memory requirement for vector data depends on the complexity of the objects. Simple polygons require less storage space.

To compare the storage space required by both models, consider Fig. 1. To represent using raster data, it will take 81 pixels. However, vector data would require significantly fewer than 81 pixels to be stored.

The scale of data is an important factor that could affect the storage requirement for raster data. Since the map is divided into a grid, the scale of the data depends on the size of the cell in the grid. For example, each cell could represent either a 1×1 km area or it could represent a 10×10 km area. The storage space required depends proportionally on the level of detail. For vector data, scale does not affect the storage requirement.

Vector Data and Data Source

The data source sometimes determines the type of data model to be used. For instance, the data obtained from remote-sensing satellites is typically in raster data format. Since the conversion of this data to vector data is time consuming, raster data model could be used in such cases.

Also, since most of the images obtained are digitized, the raster data format is more natural than the vector data format. In such cases, an extra step is required to convert the data into vector format.

Vector Data and Data Quality

Vector data is supposed to represent data with higher quality. This is preferred by cartographers, who would like to see straight lines in their maps instead of the jagged lines which usually occur in raster data models due to digitization. Raster data model quality depends on the level of detailed data being stored. As described earlier, a raster data model with data at the level of 1×1 km pixel scale will have far higher quality than a data at the scale of 10×10 km per pixel. In the case of the vector data, the quality does not depend on the scale.

Vector Data and Data Structure

Vector data requires more complex data structures to be represented in the computer. For example, to represent a polygon, it would be required to have a count of the number of vertices, coordinates of each vertex, and its relative position with respect to adjacent edges.

Raster data model could be modeled with a matrix data structure. Each cell could represent a location, and different attributes could be associated to each cell. For the above example, the data structure may look as shown in Table 1.

Vector data could be implemented with a table and a linked list. In the table, the object ID, type and its corresponding attribute values could be stored, and a link to its list of points in order could be provided.

For the above example, the vector data structure may look like Table 2.

Vector Data and Object Types

Vector data is useful for objects which could be represented mathematically in terms of simpler entities, e. g., polygons, which may be represented by a series of lines. However, it may not be suitable for objects that are complex in nature. For example, polygons with islands and polygons with disjointed regions might be difficult to represent using vector data.

Vector Data, Table 1 Raster data structure for Fig. 1

Pixel	Object ID	Attribute 1	Attribute 2
1,1	None		
2,1	None		
3,1	A1		
4,1	A1		
...	...		
9,9	None		

Vector Data and Scaling

Because of certain properties of vector data, such as the topology described above, objects represented by a vector data can be scaled without any loss of quality. Since only the objects are represented by vectors, just scaling the vectors could reproduce the actual object at different scales efficiently. This makes vector data very useful to use in applications such as maps, where a requirement is often to zoom in and out at different levels.

Also, as noted earlier, a raster data model takes up more space if data is required at a detailed level.

Vector Data and Topology

One of the most important features in spatial data is topology. Topology can be defined as a relationship that is relative to two objects. An example would be two cities connected by a road. In a vector-based data model, such information is inherent in the representation itself. In this example, a node could represent a city and an arc connecting the two nodes could represent the road. Thus, by following the arc it would be easy to find the two cities. In a raster data model, each pixel would have to be scanned to find the arc and then the cities. Hence, a vector data model is more useful when operations related to topology are important.

Vector Data and Computation

Computation for a vector data model could be expensive. This is because of the object representation. Processing on vector data involves solving complex geometrical problems e. g., finding the intersection of one polygon with another, finding distance across objects, etc. This problem of computation is more evident when the data set is large.

An example can be seen in Fig. 1. To find the river closest to the patch A1, a geometrical calculation would have to be performed between the polygon formed by A1 and other objects. This geometric calculation could be complex and computationally expensive.

Since most of the visual displays work on digitized data, a vector data would require an additional step of converting vector data to digitized format.

Vector Data and Attributes

In spatial applications, it is often necessary to associate an attribute to a geographical object. Some examples would

Object ID	Object type	Attribute 1	Attribute 2	List of Points
B1	Point			(6,4)
R1	Line string			(1,1)(3,3)(3,4) (8,9)
A1	Polygon			(2,4)(2,8)(3,9) (4,9)(5,8)(5,7)

Vector Data, Table 2 Vector data structure for Fig. 1

be populations of cities, traffic on a road, and elevation at a given point on Earth, etc. Here population, traffic, and elevation are examples of attributes related to an object. Since entities such as cities, road, etc., are represented as objects, it is easy to associate them to these attributes. This makes it easy to store such associations in the database.
Taking the example shown in Fig. 1, consider the need to associate an attribute value for the region A1. Then, all the 16 pixels in the raster data model would have to be associated with the value. However, in the case of the vector data, only object A1 could be associated to the value.

Vector Data and Applications

It is easy to write applications for vector data because the data is already represented as objects. If raster data were used, end users would have to deal with low-level details or would have to be provided with an interface which converts the data from raster data model to objects, and vice versa.
It is easy to write operations on these objects. For example, to find the area of the region A1, just use the geometric properties of the polygons to find the area of the polygon.

Vector Data and Data Modeling

Since vector data are represented as objects, it might be easy to convert logical models to a physical model. This will make it easy to design the database model. Vector data is easy to store data in an object-oriented database system. The basic building blocks of a vector data, such as points, could be stored as an object.

Vector Data and Spatial Networks

Vector data are apt for spatial networks. Spatial networks involve calculations such as the shortest path between two points and nearest neighbors. Since these operations are mainly graph-related, they could be performed easily if the objects are in the form of nodes and arcs between the nodes. Vector data satisfies this requirement. Hence, they are preferred for modeling spatial networks.

Key Applications

Generally, the vector data format could be used in any application which has spatial data. Some of the key applications of vector data are listed below:

Computer Graphics and Animation

Vector data is used in computer graphics to represent images. The quality of vector data representation of images is supposed to be better than using the traditional bitmap format. The quality of vector data does not degrade with scaling, which makes it a better choice for images such as logos, which are resized frequently. Images in vector data format are also preferred because it takes less storage space.
Most of the 3D animation uses vector data because of properties such as less storage space, and good quality.

Geographic Information Systems

Geographic information systems are computer systems that are used for creating and managing geospatial data useful for applications such as land management. Most of these applications require associating attributes to spatial data, e. g., relating population to a city. Since it is easy to associate attributes to spatial objects in a vector data format, geographic information systems prefer vector data for such cases.

Transportation Networks

In transportation networks, it is important to maintain spatially relative information between two objects. For example, the direction of a turn is spatially relative between two streets. Such information is easy to represent using vector data.

Cartography

Maps can be accurately drawn using vector data. Many cartographers prefer vector data because they can produce images without any distortion.

Future Directions

The vector data model continues to be one of the important models to represent spatial data. Vector data is more suited to discrete entities that require maintainance of topological information. Object-oriented representation of data makes vector data easy to model in a database system and applications.

Cross References

► Knowledge Representation, Spatial
► Oracle Spatial, Geometries
► Raster Data
► Road Maps, Digital
► Vague Spatial Data Types

Recommended Reading

1. Kersting, O., Dollner, J.: Interactive 3D visualization of vector data in GIS. In: Proceedings of the 10th ACM International Sym-

posium on Advances in Geographical Information Systems, pp. 107–112. McLean, Virginia (2002)
2. Khedker, U., Dhamdhere, D.M.: A generalized theory of bit vector data flow analysis. ACM Trans. Programming Languages Syst. **16**, 1472–1511 (1994)
3. Haralick, R.M.: A spatial data structure for geographic information systems. In: Freeman, H., Pieroni, G.G. (eds.) Map Data Processing, Academic, New York (1980)
4. Wu, Y.: Raster, vector, and automated raster-to-vector conversion. In: Kenney, A.R., Rieger, O.Y. (eds.) Moving Theory into Practice: Digital Imaging for Libraries and Archives, Research Libraries Group (2000)
5. Shekhar, S., Chawla, S.: Spatial Databases: A Tour

Vector Graphics

► Vector Data

Vector Graphics for the Web

► Scalable Vector Graphics (SVG)

Vector Models

► Data Models in Commercial GIS Systems

Vector (TVP)

► Spatial Data Transfer Standard (SDTS)

Vehicle Routing Problem

► Routing Vehicles, Algorithms

Vehicle Tracking Data

► Floating Car Data

Version Managed Datastore

► Smallworld Software Suite

Version Management

► Smallworld Software Suite

Versioned B-Tree

► Smallworld Software Suite

Vertical Conflation

► Conflation of Features

Virtualization, Resource

► Cyberinfrastructure for Spatial Data Integration

Visual Continuity

► Visual Momentum

Visual Data Exploration

► Exploratory Visualization

Visual Data Mining

► Image Mining, Spatial

Visual Momentum

ZHEN WEN
IBM, T. J. Watson Research Center,
Hawthorne, NY, USA

Synonyms

Visual continuity

Definition

In human computer interaction, the concept of visual momentum is used to measure how visual scene transitions impact the users ability in extracting and integrating information across multiple displays.

Main Text

The concept of *Visual Momentum* was first introduced in perception and cinematography. It refers to the impact of a transition from one scene to another on the cognitive processes of an observer. Particularly visual momentum is used to describe the impact on the observer's ability to extract task relevant information. The concept was later used in the area of Human Computer Interaction (HCI). In HCI, the amount of visual momentum supported by a display system is inversely proportional to the mental effort required to integrate a new display into the context of the overall information space and the user's information needs.

Cross References

► Information Presentation, Dynamic

Visual Thinking

► Exploratory Visualization

Visualization of Constraint Databases

► Visualization of Spatial Constraint Databases

Visualization of Spatial Constraint Databases

SHASHA WU
Department of Math, Physics, and Computer Science, Spring Arbor University, Spring Arbor, MI, USA

Synonyms

Visualization of constraint databases; Visualization of spatiotemporal constraint data; Static displays in constraint databases

Definition

The visualization of spatial constraint databases is an application that reads spatial data from spatial constraint databases, understands the meaning of the data based on the spatial data models, and displays those data as computer graphics to help cognition, analyzing and reasoning.

Main Text

To display the spatial data in a constraint database system, the system will first transform the vector-based spatial data, which is represented by linear equality and linear inequality constraints in a disjunctive normal form (DNF), into a set of points, segments, and convex polygons, then associate a color with the individual shapes and project them onto the output devices.

For example, the following constraint tuple represents a triangle. The visualization of spatial constraint databases can display the triangle on a computer through above process in three steps.

$$ABC(x,\ y) \quad :- \quad x \geq 1,\ y \geq 1,\ x + y < 4\,.$$

First, convert the three linear inequality constraints in the tuple into three linear equations as follows:

$$x = 1$$
$$y = 1$$
$$x + y = 4\,.$$

Second, compute the set of intersection points for each pair of equations. Remove any points that do not satisfy the inequality constraints in the original tuple so that only the corner points of the polygon will be kept in the set. Order the resulting corner points in a sequence so that each pair of adjacent points represent an edge of the polygon. For example, the output point sequence for above problem is ((1,1), (1,3), (3,1)).

Finally, call the operating system's polygon function to draw the triangle based on the resulting sequence of corner points as well as the color assigned to relation *ABC*.

The above general visualization process was implemented in most constraint database systems to display 2-D spatial constraint data. It was also used to animate 2-D spatiotemporal constraint data. Theoretically, it can be applied to visualize 3-D or higher dimensional data for constraint databases.

Cross References

► Constraint Database Queries
► Constraint Databases and Data Interpolation
► Constraint Databases and Moving Objects
► Constraint Databases, Spatial
► Linear Versus Polynomial Constraint Databases
► MLPQ Spatial Constraint Database System
► Raster Data
► Vector Data

References

1. Revesz, P.: Introduction to Constraint Databases. Springer, New York (2002)
2. Rigaux, P., Scholl, M., Segoufin, L., Grumbach, S.: Building a Constraint-Based Spatial Database System: Model, Languages, and Implementation. Inf. Syst. **28**(6), 563–595 (2003)

Visualization of Spatio-temporal Constraint Data

► Visualization of Spatial Constraint Databases

Visualization, Photorealistic and Non-photorealistic

JÜRGEN DÖLLNER
Hasso-Plattner-Institute at the University of Potsdam, Potsdam, Germany

Synonyms

NPR; Non-photorealistic computer graphics; Non-photorealistic rendering

Definition

Non-photorealistic rendering denotes image synthesis techniques that apply illustrative, stylistic, and artistic depiction techniques to generate visual representations that partially or completely dismiss photorealism in favor of abstraction. With non-photorealistic rendering cartography and GIS obtain an innovative technology for designing and implementing of user and task oriented visualization systems and applications; it facilitates the implementation of visualization techniques that take into consideration, for example, cartographic design principles and specifics of geoinformation display.

Terminology

Since around 1990, NPR has established itself as a key category and discipline in computer graphics. Most researchers agree that the term "non-photorealistic" is not satisfactory because the notion of realism itself is not clearly defined nor its complement, the non-photorealism.

Historical Background

With the advent of NPR a repertoire of illustrative, expressive, and artistic computer graphics techniques offer different ways of presenting and communicating visual information compared to representations achieved by traditional computer graphics, which has been strongly aligned to photorealistic rendering. In general, NPR enables developers to present and depict visual information in a purpose-oriented and task-oriented way using principles of classical illustration techniques. In terms of computer graphics, photorealistic and non-photorealistic rendering primarily differ in the way of, e. g., shape representation, coloring, lighting, shading, and shadowing.
NPR offers extensive control over expressivity, clarity, and aesthetics, thereby the resulting pictures "can be more effective at conveying information, more expressive or more beautiful"[1]. In particular, the process of image synthesis is not strictly defined by principles of physical light emission in contrast to ray-tracing, radiosity, or photon-tracing approaches used in photorealistic rendering. Strothotte and Schlechtweg [2] as well as Gooch and Gooch [3] give a broad introduction to concepts and algorithms of non-photorealistic computer graphics.
In NPR, the wealth of techniques comprise stylized digital half toning for simulating handcrafted depictions, such as stippling, hatching, or engraving conveying illumination, curvature, texture, and tone in an image [4]. Furthermore, techniques exist for generating technical illustrations automatically [5] or for reproducing pencil or pen-and-ink drawings or sketchy depictions.
NPR turns out to be an enabling technology for designing and implementing effective visualization systems overcoming the traditional mindset established by photorealistic computer graphics. In the scope of geoinformation, NPR facilitates the implementation of abstract, aesthetically pleasant, and adequate techniques to communicate geoinformation by means of visual digital media, e. g., illustrative city model depictions.

Scientific Fundamentals

Elements of Non-photorealistic Rendering

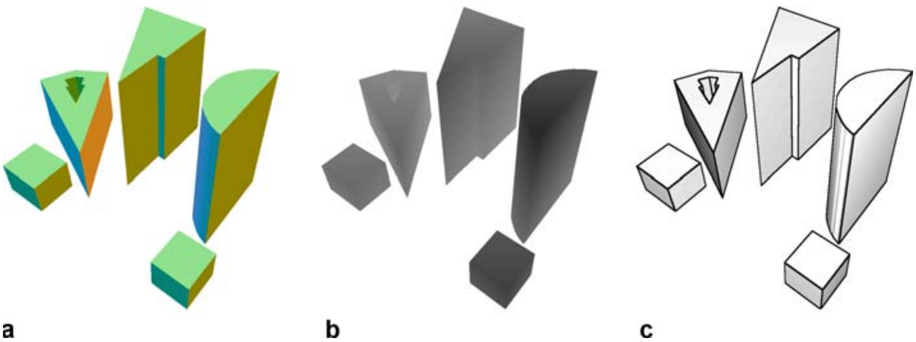
NPR aims at the expressive visualization of relevant information abstracting from the physical geometry and appearance of the underlying objects and their appearance. Abstraction represents one of the key characteristic elements in NPR because it allows the user to get insight in and understanding of complex objects, structures, and relations. To achieve abstraction, NPR uses edge enhancement, illumination and shading techniques different to those found in photorealistic rendering.

Edge Enhancement

Edge enhancement represents a key element in NPR techniques. "A few good lines" [6] are generally sufficient to achieve expressive depictions of complex geometry using the human viewing ability to complete incomplete representations. For NPR, enhancement techniques need to define edge detection and edge visualization. Two principle techniques are distinguished: image-space edge enhancement and object-space edge enhancement [7].
Object-space edge enhancement (Fig. 1) determines visually important edges for each individual object based on its geometry and semantics [8,9,10]. Identified edges are enhanced using additional 3D geometry, which is drawn

V

Visualization, Photorealistic and Non-photorealistic, Figure 1 Examples of object-space edge enhancement. **a** Original 3D scene. **b** Curved-based edge enhancement. **c** Sketchy edge enhancement



Visualization, Photorealistic and Non-photorealistic, Figure 2 Examples of object-space edge enhancement. **a** Contents of the normal G-buffer. **b** Contents of the depth G-buffer. **c** Corresponding scene with superimposed edges map derived from normal and depth G-buffers

using strokes captured by textures and applied to the additional scene geometry. This allows designers to define different types of strokes, brushes, and line styles.

Image-space edge enhancement (Fig. 2) identifies visually important edges for a given image in image-space [11]. Typically, these techniques use non-visible auxiliary image buffers, the G-buffers [12], to determine edges caused by silhouettes, strong curvatures, or object borders. G-buffers store geometric information of the visible scene geometry on a per-pixel basis, e. g., normal vectors, depth values, or object identity. Visually important edges can be calculated using image processing. For example, image operations that calculate discontinuities in the depth buffer detect object silhouettes, whereas discontinuities in the normal buffer detect edges within object surfaces. To visualize detected edges, the edge-map approach merges the contents of G-buffers and superimposes the resulting 2D edge map on top of the scene depiction [13].

Non-photorealistic Illumination

Non-photorealistic illumination includes techniques that illuminate and shade 3D objects and integrate depth cues in the depiction; they do not necessarily follow principles of physical light distribution and interaction. A first NPR illumination model was introduced by Gooch and Gooch [5], "where the lighting model uses both luminance and changes in hue to indicate surface orientation, reserving extreme lights and darks for edge lines and highlights. The lighting model allows shading to occur only in mid-tones so that edge lines and highlights remain visually prominent."

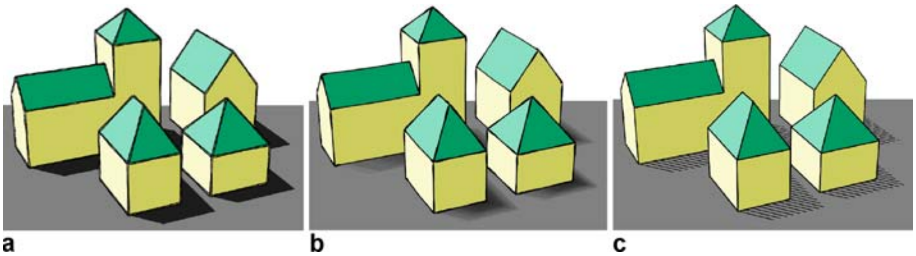
Shadows represent another important category of elements of NPR illumination (Fig. 3). They allow the precise depth perception of depicted scene geometry and their spatial relations. For example, the length of a projected shadow gives the observer a hint about the object size and its relative position within the scene. In NPR, both hard and soft shadows are considered. In contrast to photorealistic rendering, the goal is not to determine the shadow position and shape as physically correct as possible but to instrumentalize the shadow to optimize image understanding. The physical correctness plays a less important role for this function [14].

Non-photorealistic Shading

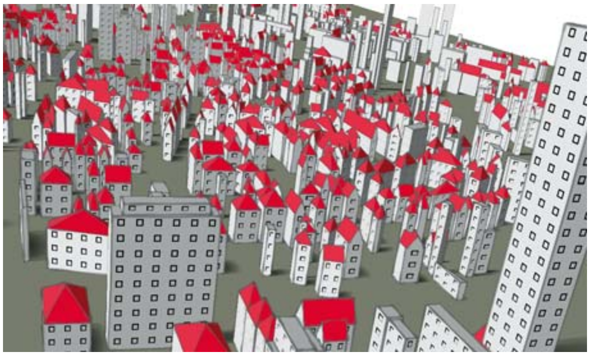
In photorealistic rendering surfaces are commonly shaded by interpolating illumination intensity [15] between vertices (Gouraud shading) or calculating the intensity on a per-pixel basis (Phong shading [16]). In contrast, NPR uses different approaches to calculate the shades and to render them across surfaces.

To calculate shades, the variety of shades and used colors is typically reduced in illustrations (Fig. 4). In general, only a few tones (e. g., 3 or 4) are used for a given color hue. This n-tone shading achieves a cartoon-like impression [8]. To further enable a depth impression, objects far away are rendered using de-saturated tones.

To visualize calculated shades, NPR applies different rendering techniques, such as hatching. It uses parallel, dense lines that follow the principle curvature of a given surface [17]. The hatched surface expresses the illumination as well as material properties. Hatching-based and stroke-



Visualization, Photorealistic and Non-photorealistic, Figure 3 Three different types of NPR shadows. **a** Hard shadows, **b** Soft shadows, and **c** Hatched shadows



Visualization, Photorealistic and Non-photorealistic, Figure 4 Example of n-tone-shading applied to a set of building models. Only three variants of two principle colors (*gray, red*) are used

based approaches for NPR shading can be implemented on modern computer graphics hardware even in real-time [4].

Photorealistic vs. Non-photorealistic Rendering

The concepts of photorealistic rendering still shape principle techniques used for visualizing 3D geoinformation, e. g., by geo-virtual 3D environments.

In the scope of virtual reality (VR), a large collection of algorithms and data structures provide efficient rendering of large, complex 3D terrain data and heterogeneous geo-spatial objects, e. g., optimizing the rendering of digital terrain models or large collections of heterogeneous 3D objects. These methods are focusing on efficient rendering and require high quality of texture and geometry data. Established photorealistic rendering techniques include ray-tracing, photon-tracing, and radiosity; today's high-end commercial rendering systems use hybrid algorithms that combine these approaches.

Numerous optimizations for virtual environments have been developed, such as discrete and continuous multiresolution representations of geometric objects, to enable level-of-detail (LOD) rendering and data streaming, view-frustum culling, occlusion culling, billboard clouds and impostor techniques to reduce the geometric complexity of scenes, scene-graph parallelization and optimizations to take full advantage of the graphics pipeline, and recently out-of-core visualization techniques, which directly render large amounts of data from external memory in real-time. Akenine and Haines [18] provide a general introduction to these subjects.

While research in virtual reality provides efficient solutions for handling complex 3D scenes, it does not provide alternatives to the physically oriented illumination models and graphics designs. For example, the Phong illumination model [16], which approximates the physical interaction of light in 3D environments, is actually shaping current visualizations – it was not possible until recently to develop alternatives if real-time rendering was required because it forms part of the hardware-implemented rendering pipeline. This situation changed fundamentally with the advent of programmable computer graphics hardware.

Another limitation represents single-pass rendering. The rendering engine traverses the scene description in a single pass to generate the image and directly sends graphics data to the graphics hardware. However, a large number of graphics designs become possible only by more complex, multiple rendering passes, which collect data, prepare intermediate images and can perform imaging operations.

Key Applications

Applications of NPR in GIS

Looking at graphics quality achieved in 3D cartographic applications or GIS, inherent limitations of photorealistic techniques become apparent: Photorealistic rendering techniques require and, therefore, depend on geometric and graphical detail to achieve impressive results. For example, a 3D city model without high-resolution facade textures typically remains an inconsistent visual artifact. The lack of sufficient detail is often frustrating – not only because highly detailed data would actually be required at early stages of construction but also because without such detail the visual results are not convincing from a perception's and an observer's point of view. Fig. 5 illustrates this phenomenon, showing both photorealistic and non-photorealistic depictions of the same 3D city model.

Non-photorealistic computer graphics offer a large potential for developing specialized visualization techniques in

Visualization, Photorealistic and Non-photorealistic, Figure 5 Comparison of near-photorealistic and non-photorealistic depictions of the same 3D scene

multimedia cartography and 3D GIS such as for illustrative, artistic, and informal information display. Most importantly, non-photorealism provides excellent means for visual abstraction as a primary technique to effectively communicate complex spatial information and not only allows us to implement traditional cartographic visualization techniques as part of interactive visualization applications, but also offers an excellent foundation for developing new cartographic concepts for visualizing geoinformation such as illustrative city models and landscape models. NPR also raises several questions with respect to its impact to map-like and cartographic presentations. What are specific NPR techniques for interactive map-based depictions? How do non-photorealistic geovisualizations cooperate with photorealistic ones? How can one combine them? What are the specific elements of photorealism and non-photorealism in the human visual processing? How does one take advantage of these abilities for a more effective visual communication?

Example: Illustrative City Models

In cartography, the visual representation of city models has a long history and has yielded many principles for drawings of this category. The most prominent examples include panoramic maps of cities and landscapes as well as bird's-eye views of cities. They show a high degree of visual clarity and geometric abstraction – preferring abstraction to realism.

City model illustrations can be considered as map-related 3D representations of urban geodata and georeferenced thematic data. In addition to 2D geodata, such as topographic maps, thematic maps, and 2D ground plans, they include 3D geo-data, for example, 3D terrain models, 3D building models, and 3D vegetation models.

In classical bird's-eye views both orthogonal and perspective projections are used. The city depictions of Merian, the Bollmann's maps of many European as well as US cities are well-known examples of 'pictorial city maps'. Traditionally they choose the east or west direction for projection because important, optically dominant landmarks, such as well known public buildings and churches, are oriented that way. In addition, map designers often attempt to diagonally capture important quarters and streets of houses. Whatever direction is taken, approximately half of the model is not shown in static views. Maps of city models attempt to maximize visual clarity. Most importantly, edges are enhanced to stress the contours of buildings. Colors are chosen according to semantics and aesthetics – they do not necessarily correspond to the natural colors. The overall number of colors is kept small.

The terrain model represents the reference surface for city models. In traditional city maps, the terrain surface is often ignored or just indicated where its morphology is significant. To cope with complex geometry, detail reduction is used both at the technical level by multi-resolution modeling and level-of-detail techniques, and at the semantic level by generalizing buildings to quarters if their distance to the camera exceeds a given threshold. As the sides of buildings provide more characteristic information about them than the roofs, the roofs can be scaled down in height while the main bodies of the buildings are scaled up. This is done in traditional panoramic views of cities to meet the spectator's expectation, who is used to see the city as a pedestrian.

In a complex and dense urban area landmarks serve as means of orientation for the user. Landmarks consist mainly of those buildings that are known a priori to the user – typically monuments and buildings of public interest. Therefore, they require an exact 3D graphics model. The visualization should guarantee that landmarks, even if they are nearly out of the view volume, do not disappear. The multi-resolution mechanism might want to treat

Visualization, Photorealistic and Non-photorealistic, Figure 6 Example of a non-photorealistic depiction of a virtual 3D city model based on real-time rendering techniques for image-space edge enhancement, shadowing, and n-tone shading

them differently from the remaining buildings, because the eye is sensitive to even small changes of well-known objects.

Figure 6 exemplifies a visualization technique that achieves expressive illustrations of large-scale 3D city models [19]. It defines a collection of city model components and has been implemented by a real-time multi-pass rendering algorithm. It is based on image-space edge enhancement, color-based and shadow-based depth cues, and procedural texturing. The technique shows that illustrative visualization provides an effective visual interface to urban spatial information and associated thematic information, offering many degrees of freedom for graphics design. Primary application areas include city and landscape planning, cartoon worlds in computer games, and tourist information systems.

Future Directions

Implementation

Today's 3D computer graphics feature a partially programmable rendering pipeline and offers efficient parallel computing power. Programmable computer graphics hardware represents the prerequisite for advanced real-time rendering algorithms both in the fields of photorealism and non-photorealism.

Graphics programming interfaces (e. g., OpenGL or DirectX) offer shading languages that allow developers to formulate vertex and fragment programs used in the rendering pipeline by a high-level programming language (e. g., OpenGL Shading Language OpenGLSL [20]). Generally, multiple rendering passes and a variety of object-specific and effect-specific shaders are used to synthesize a single image.

Recommended Reading

1. Durand, F.: An invitation to discuss computer fepiction. In symposium on non-photorealistic animation and rendering (NPAR) (2002)
2. Strothotte, T., Schlechtweg, S.: Non-Photorealistic Computer Graphics: Modeling, Rendering and Animation. Morgan Kaufman, San Francisco (2002)
3. Gooch A., Gooch B.: Non-Photorealistic Rendering. A K Peters Ltd, Wellesley, MA (2001)
4. Praun, E., Hoppe, H., Webb, M., Finkelstein, A.: Real-time hatching. In: Computer Graphics (Proceedings of ACM SIGGRAPH 2001), pp. 581–586 (2001)
5. Gooch, A., Gooch, B., Shirley, P., Cohen, E.:. A non-photorealistic lighting model for automatic technical illustration. Computer Graphics (Proceedings of SIGGRAPH 1998), pp. 447–452 (1998)
6. Sousa, M.C., Prusinkiewicz, P.: A few good lines: suggestive drawing of 3D models. In: Proceedings of Eurographics, Computer Graphics Forum, 22(3):381–390 (2003)
7. Isenberg, T., Halper, N., Strothotte, T.: Stylizing silhouettes at interactive rates: from silhouette edges to silhouette strokes, In: Computer Graphics Forum (Proceedings of Eurographics), 21(3):249–258 (2002)
8. Döllner, J., Walther, M.: Real-time expressive rendering of city models. In: Seventh International Conference on Information Visualization, pp. 245–251, London (2003)
9. Isenberg, T., Freudenberg, B., Halper, N., Schlechtweg, S., Strothotte T.: A developer's guide to silhouette algorithms for polygonal models. IEEE Computer Graphics and Applications, 23(4):28–37, (2003)
10. Northrup, J.D., Markosian L.: Artistic silhouettes: A hybrid approach. In: Proceedings of NPAR (2000) 2000
11. Mitchell, J., Brennan, C., Card, D.: Real-time image-space outlining for non-photorealistic rendering. ACM SIGGRAPH 2002 Sketch (2002)
12. Saito T., Takahashi T.: Comprehensible Rendering of 3-D Shapes. Computer Graphics (Proceedings of ACM SIGGRAPH '90), pp. 197–206 (1990)
13. Nienhaus, M., Döllner, J.: Edge-enhancement – an algorithm for real-time non-photorealistic rendering. Journal of WSCG, 11(2):346–353 (2003)
14. Ware, C.: Information Visualization. Morgan Kaufman, San Francisco (2000)
15. Gouraud, H.: Continuous shading of curved surfaces. In Rosalee Wolfe (ed.), Seminal Graphics: Pioneering efforts that shaped the field. ACM Press, New York (1998)
16. Phong, B.T.: Illumination for Computer Generated Images, Comm, vol 18(6):311–317. ACM, New York (1975)
17. Buchin, K., Sousa, M.C., Döllner, J., Samavati F., Walther, M.: Illustrating terrain using direction of slope and lighting. In: 4th ICA Mountain Cartography Workshop (2004)
18. Akenine-Möller, T., Haines, E.: Real-Time Rendering, A K Peters Ltd, Wellesley, MA (2002)
19. Döllner, J., Buchholz, H., Nienhaus, M., Kirsch, F.: Illustrative visualization of 3D city models. In: Proceedings of Visualization and Data Analysis 2005 (Electronic Imaging 2005, SPIE Proceedings), pp. 42–51 (2005)

20. Rost, R.J.: OpenGL Shading Language, 2nd edn. Addison Wesley, Boston (2006)

Visualization Tool

► GRASS

Visualizing Constraint Data

SHASHA WU
Department of Math, Physics, and Computer Science, Spring Arbor University, Spring Arbor, MI, USA

Synonyms

Constraint database visualization; Isometric color bands displays

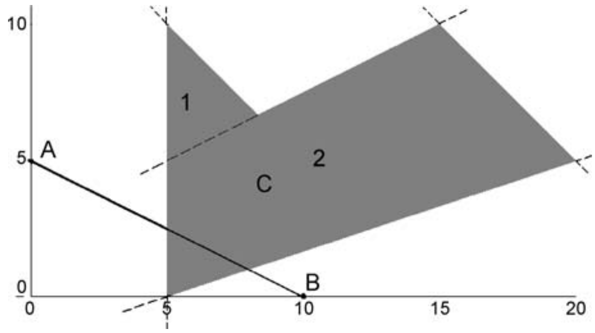
Definition

In general, visualization is any technique for creating images, diagrams, or animations in order to present any message. Scientific visualization is an application of computer graphics which is concerned with the presentation of potentially huge quantities of laboratory, simulation or abstract data to aid cognition, hypotheses building and reasoning.

In a spatial database system, spatial information is usually stored in the format of *raster data* or *vector data*. To visualize *raster data*, the visualization application has to convert the geographical information associated with each pixel into a specific color and present each pixel individually. To visualize *vector data*, the visualization application has to identify the geometrical primitives such as points, lines, curves, and polygons, convert the original geospatial coordinate system to screen coordinate system, associate a particular color to each shape, and then output those shapes through the drawing functions provided by the operating system.

The geometrical primitives used by vector data are all based upon mathematical equations to represent images in computer graphics. The constraint databases use linear equality and inequality constraints as its primitive data type to represent spatial data. That makes constraint databases a natural solution for storing, retrieving and displaying vector-based spatial data.

The visualization of spatial data in a constraint database system is the process of transforming the vector-based spatial data, which is represented by linear equality and linear inequality constraints in a disjunctive normal form (DNF) [9,13], into a set of points, segments, and convex polygons, and associating a color with the individual shapes, and then projecting those shapes onto the output devices.



Visualizing Constraint Data, Figure 1 Visualization of point, polyline and polygon in constraint databases

For example, Fig. 1 is the visualization result of following three linear constraint relations:

$$\begin{aligned} &PointA(x,y) &&:- \quad x=0,\ y=5.\\ &LineAB(x,y) &&:- \quad x\ge 0,\ y\ge 0,\ x+2y=10.\\ &PolygonC(i,x,y) &&:- \quad i=1,\ x-2y\le -5,\ x+y\le 15,\\ & && \qquad -x\le -5.\\ &PolygonC(i,x,y) &&:- \quad i=2,\ -x+2y\le 5,\ x+y\le 25,\\ & && \qquad x-3y\le 5, -x\le -5. \end{aligned}$$

Constraint databases are well-suited for animation because they allow any granularity for the animation without requiring much data storage [9]. Beyond that, the ability of representing spatiotemporal and non-spatiotemporal data in an identical format and the support of recursive queries make constraint databases a good approach for many difficult visualization problems, such as the visualization of recursively-defined spatiotemporal concepts discussed in [11,12].

Although most existing constraint database systems can only visualize 2-D spatiotemporal objects, they can be extended to visualize three or even higher dimensional spatiotemporal objects. By introducing new variables into the linear constraints, constraint databases can represent higher dimensional objects similar to 2-D objects. The visualization of those objects is reduced to a process of visualizing the union of basic higher dimensional blocks.

Historical Background

Constraint databases, including spatial constraint databases, were proposed by Kanellakis, Kuper and Revesz in 1990 [5]. They showed in [6] that "efficient, declarative database programming can be combined with efficient constraint solving" and suggested that the Constraint

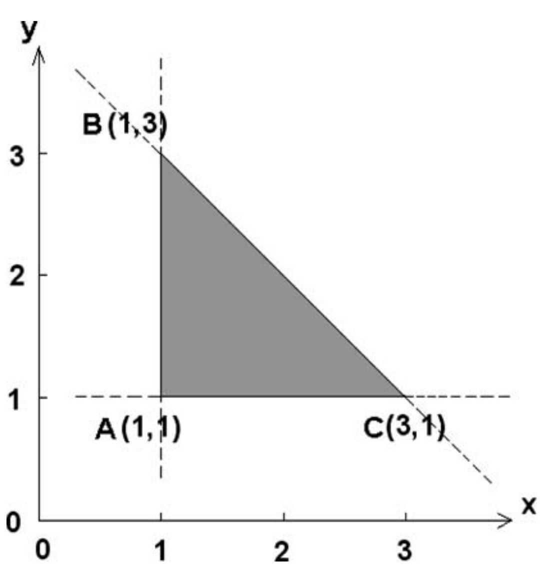
Database framework can be applied to manage spatial data. A few years later, several spatial constraint databases systems, such as the MLPQ system [10], the CCUBE system [1], the DEDALE system [4], and the CQA/CDB system [3], were developed. During the development of those systems, convex polygons were the major visualization blocks presenting the outputs of the spatial constraint databases systems. Extreme Point Data Models like the Rectangles Data Model [9] and Worboys' Data Model are also implemented in some constraint database systems. For example, the MLPQ system implements both the regular polygon visualization and the Parametric Rectangle Visualization. The last one, named the PReSTO system, implements several special animation features like *Collide* and *Block*. With the increased number of applications developed from the spatial constraint database systems, the aim of efficiently and naturally visualizing sophisticated spatial or spatiotemporal constraint data attracts more and more attention.

Scientific Fundamentals

Static Displays

Any 2-D static display can be reduced to the visualization of points, polylines, and polygons. In constraint databases, a point can be directly represented by linear equations over two variables (x, y). For example, point $A(1,1)$ in Fig. 2 can be represented as:

$$A(x,y) \quad :- \quad x = 1,\ y = 1\,.$$



Visualizing Constraint Data, Figure 2 Representing spatial object by convex polygon(s)

It is a trivial problem to visualize a point with x and y coordinations. Things are a little bit more complex for polylines and polygons.

The line segment between points $B(1,3)$ and $C(3,1)$ can be represented as:

$$BC(x,y) \quad :- \quad x + y = 4,\ x \geq 1,\ x \leq 3,\ y \geq 1,\ y \leq 3\,.$$

A polygon can be either a convex polygon or non-convex polygon. A convex polygon can be directly represented by a set of conjunctive linear inequality constraints over the two variables(x, y). A non-convex polygon must be first partitioned into convex components. Then, it can be represented by constraint databases and visualized through the union of the convex components.

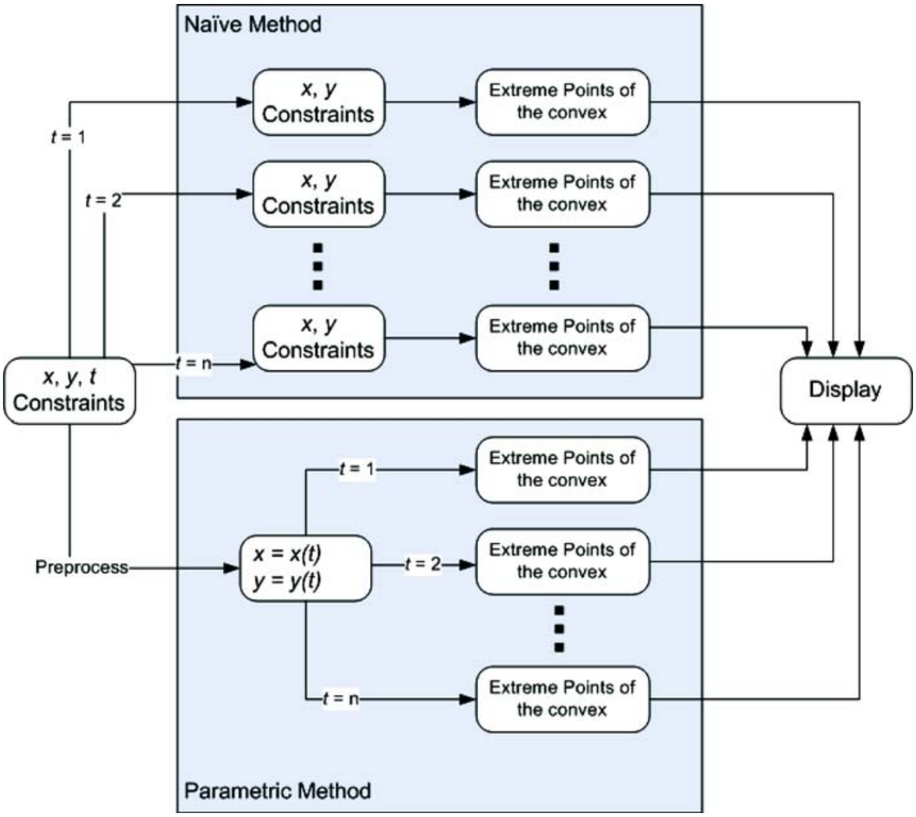
This way, any vector data can be represented by disjunctive normal form formulas of linear equations and linear inequality constraints over x and y [9]. For example, the triangle formed by vertices $A(1,1)$, $B(1,3)$, and $C(3,1)$ can be represented by the following conjunction of linear inequality constraints:

$$ABC(x,\ y) \quad :- \quad x \geq 1,\ y \geq 1,\ x + y \leq 4\,.$$

Most of the original spatial objects are described by polygons. To represent those objects, constraint databases have to first decompose polygons into convex components. For some kind of polygons, it is a NP-hard problem [8]. But for most of commonly used polygons, it is possible to find polynomial polygon partition algorithms [2,7,14]. Among all of them, the simplest algorithm is polygon triangulation, which represents a polygon through a set of triangles. However, triangulation results in a large, sometimes prohibitive, number of convex components in the partition. Given a polygon with n vertices, the number of triangles in the partition is $n-2$. To solve the problem, Keil [7] proposed an algorithm that can generate an optimal number of convex components in the partition for most types of polygons. However, the time complexity of his algorithm is $O(N^2 n \log n)$. To reduce the time complexity, a less optimal algorithm is proposed and implemented in [13], in which the polygon is first triangulated and then the adjacent triangles are merged to reduce the number of convex components in the result.

Animation of Spatiotemporal Objects

Each *spatiotemporal object* has a *spatial extent* and a *temporal extent*. The spatial extent represents the set of points in space that belong to the object. The temporal extent represents the set of time instances when the object exists. The shape and the location of a 2-D spatiotemporal object may change over time. In constraint databases, each 2-D



Visualizing Constraint Data, Figure 3 Naive and parametric animation methods (see Fig. 16.11 in [9])

spatiotemporal object is represented by linear constraints over the three variables (x, y, t) in disjunctive normal form formula [9].

There are two different animation methods, as shown in Fig. 3, to visualize the spatiotemporal constraints:

The *naive animation method* works directly on constraint databases. It finds the linear constraint relations that have only two spatial variables named as x and y to represent the extreme points of the polygon for each time instance t_i, by instantiating the variable t to t_i in the original linear constraint tuple. Then it calls the graphic library functions provided by the operating system to output the polygon. The whole computation will be executed every time the user requests an animation display. It is a time consuming process and often times causes many delays and jumps in the animation.

The *parametric animation method* has a preprocessing step and a display step to speed up the animation. The preprocessing step is executed at the time the constraint relation is loaded or constructed. It first computes the *extreme points* of each polygon based on its constraint tuple. Then, each polygon is describable by a sequence of extreme points. Finally, each extreme point is represented by parametric functions $x = x(t)$ and $y = y(t)$, which will be kept in memory until the close of the constraint relation. The display step is executed every time the user requests an animation display. After the user specifies the range and the granularity of time for the animation and sends the request to the system, the extreme point parametric functions are loaded and the time variable t is instantiated several times based on the required granularity. It generates a sequence of polygon outputs and sequentially and smoothly outputs them onto the monitor. This way, the spatiotemporal data are visualized as an animation display.

Key Applications

The visualization of spatial constraint databases are similar to the visualization of other GIS systems, such as the ARC/GIS system. However, the power of efficiently describing infinite spatial and spatiotemporal data and the support of recursive queries make the visualization of spatial constraint databases more attractive for complex problems like visualization of the recursively defined spatiotemporal concepts [11]. These applications typically include problems where various kinds of spatial and spatiotemporal information such as maps, population, meteorology phenomena, and moving objects are represented and visualized. The following are some examples of such applications.

Visualization Functions of the MLPQ Constraint Database System

The MLPQ Constraint Database System implemented many visualization operators. For example, the *Complement* operator returns the complement of the given spatial object. The *Difference* operator generates the difference between two spatial objects. Three commonly used visualization operators are described as follows.

2D Animation: The MLPQ Constraint Database system can display the spatiotemporal relations in animations. It provides a set of buttons for the user to control the displaying of the animation. The animation button allows the user to set the start and end time, the time interval of two frames, and the speed of the animation. The play and play back buttons play the animation forward and backwards, respectively. The first, forward, next, and last buttons allow the user to navigate between frames.

Block: Some spatial objects like light and fire are formed by a set of independent points. If some of the points are blocked by the presence of another object, the rest of the points just continue moving along the trajectory determined by the transformation function. The block operator is designed to visualize such situations in the constraint databases. It takes two relations and a time instance t_k as the inputs and returns a new relation that represents the points of the first relation at time instance t_k that are not blocked by the second relation at any time before t_k. Based on the block operator, people can easily visualize the spatial objects such as the shadow of a ball or show how a lake can block the movement of a fire in the forest.

Collide: A common scenario between moving objects is the collision. The *Collide* operator is designed to visualize the collision situation in animation. It assumes an extra attribute for spatiotemporal objects called mass. Suppose there are two objects that do not change their shape and that are traveling at uniform speed defined by the transition function in the constraint databases. The *Collide* operator will generate a new relation that expresses the motion of the objects before and after collision. This operator can be used to visualize applications like the crash of two cars or the contact of two billiard balls.

Applications Based on Recursively Defined Concepts

Visualization of recursively defined concepts is a general problem that appears in many areas. For example, drought areas based on the Standardized Precipitation Index (SPI) and long-term air pollution areas based on safe and critical level standards are recursively defined concepts. In [11], a general and efficient representation and visualization method was proposed to display recursively defined spatio-temporal concepts. Sample applications such as visualization of drought and pollution areas were implemented to illustrate the method.

Applications for Epidemiology

Efficient computerized reasoning about epidemics is important to public health and national security, but it is a difficult task because epidemiological data are usually spatiotemporal, recursive, and fast changing, hence hard to handle in traditional relational databases and geographic information systems. In [12], a particular epidemiological system called WeNiVIS was implemented based on the visualization of spatiotemporal constraint databases. It enables the visual tracking of the West Nile Virus epidemic in Pennsylvania and helps people to predicate the high risk areas.

Future Directions

Spatial constraint databases provide an efficient way to store and query spatial or spatiotemporal data over Internet. There is a growing number of web-based spatial constraint database applications. Most of the applications ask for high-level visualization methods of constraint data to improve their user interfaces. Methods to enhance the performance of visualizing spatial constraint databases over Internet are being developed.

By adding one more parameter in the database, Spatial Constraint Databases can easily represent 3-dimension data. However, by the time of writing this entry, only primitive visualization methods like isometric color bands are implemented in some existing constraint database systems. For example, the map can be visualized by discrete color zones according to the values of variable z, which can be used to represent the value of elevation, precipitation, or temperature. Using 2-D images to visualize 3-D objects is a temporary solution for this problem. Compared to the 3-D visualization of other commercial GIS systems, this method has many restrictions on the 3-D objects to be visualized and the result is not impressive. The implementations of real 3-D visualization of constraint databases are being developed.

Cross References

► Constraint Database Queries
► Constraint Databases and Data Interpolation
► Constraint Databases and Moving Objects
► Constraint Databases, Spatial
► MLPQ Spatial Constraint Database System
► Raster Data
► Vector Data

V

Recommended Reading

1. Brodsky, A., Segal, V., Chen, J., Exarkhopoulo, P.: The CCUBE constraint object-oriented database system. Constraints **2**(3–4), 245–277 (1997)
2. Chazelle, B., Dobkin, D.: Decomposing a polygon into its convex parts. In: Proc. 11th Annual ACM Symposium on Theory of Computing, pp. 38–48., Atlanta, Georgia, 1979
3. Goldin, D., Kutlu, A., Song, M., Yang, F.: The Constraint Database Framework: Lessons Learned from CQA/CDB. In: Proc. International Conference on Data Engineering, pp. 735–737, Bangalore, India, 2003
4. Grumbach, S., Rigaux, P., Segoufin, L.: The {DEDALE} system for complex spatial queries. In: Proc. ACM SIGMOD International Conference on Management of Data, pp. 213–224, Seattle, Washington, 1998
5. Kanellakis, P.C., Kuper, G.M., Revesz, P.: Constraint Query Languages. In: Proc. ACM Symposium on Principles of Database Systems, pp. 299–313, Nashville, Tennessee, 1990
6. Kanellakis, P.C., Kuper, G.M., Revesz, P.: Constraint query languages. Journal of Computer and System Sciences **51**(1), 26–52 (1995)
7. Keil, J.M.: Decomposing a Polygon into Simpler Components. SIAM Journal of Computing **14**(4), 799–817 (1985)
8. O'Rourke, J., Supowit, K.J.: Some NP-hard polygon decomposition problems. IEEE Transactions on Information Theory **29**(2), 181–190 (1983)
9. Revesz, P.: Introduction to Constraint Databases. Springer, New York (2002)
10. Revesz, P., Li, Y.: MLPQ: A linear constraint database system with aggregate operators. In: Proc. 1st International Database Engineering and Applications Symposium, pp. 132–137. IEEE Press, Washington, DC (1997)
11. Revesz, P., Wu, S.: Visualization of recursively defined concepts. In: Proceedings of the 8th International Conference on Information Visualization, pp. 613–621. IEEE Press, Washington, DC (2004)
12. Revesz, P., Wu, S.: Spatiotemporal Reasoning About Epidemiological Data. Artificial Intelligence in Medicine **38**(2), 157–170 (2006)
13. Rigaux, P., Scholl, M., Segoufin, L., Grumbach, S.: Building a Constraint-Based Spatial Database System: Model, Languages, and Implementation. Information Systems **28**(6), 563–595 (2003)
14. Schachter, I.: Decomposition of polygons into convex sets. IEEE Transactions on Computers **27**(11), 1078–1082 (1978)

Voronoi Diagram

JAMES M. KANG
Department of Computer Science and Engineering,
University of Minnesota, Minneapolis, MN, USA

Synonyms

Voronoi tesselation;
Dirichlet tessellation;
Thiessen polygons

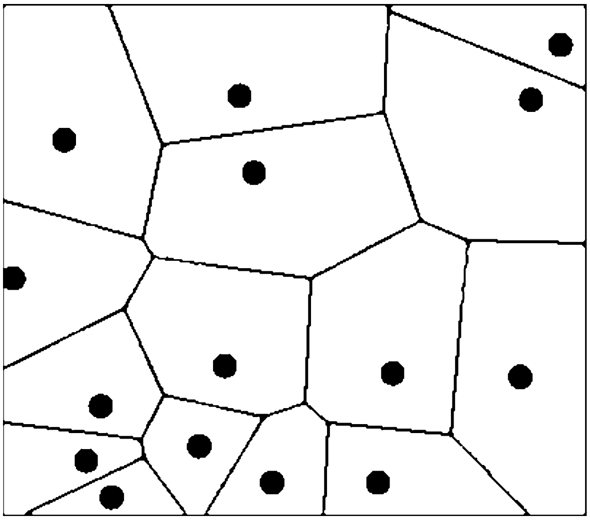
Definition

In general, this method decomposes a set of objects in a spatial space to a set of polygonal partitions. Figure 1 shows an example of a Voronoi Diagram where each object (denoted by a dot) is placed in a separate polygon. Formally, for any set of objects *o* in a two- or three-dimensional space, a polygonal shape surrounds the object such that approximately any point *p* in the polygon is closer to its generated object than any other generated object.

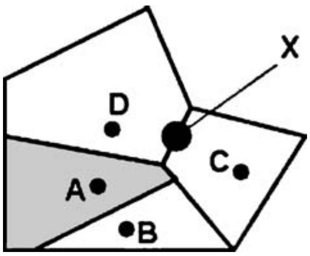
For example, Fig. 2 shows a sub-section of the Voronoi Diagram shown in Fig. 1.

Any point within the shaded region containing the generated point *A* is closer to *A* than any other generated point *B*, *C*, and *D*. However, the points on the perimeter of a polygon may be the midpoint of two points (i. e., point *X* has the same distance from *A* and to *C*). These midpoints are considered as the hyperplane that separate these two generative points. The set of hyperplanes generates the polygonal shapes around the generated objects.

This definition explains the first order of the Voronoi Diagram. There are other variants to the Voronoi Diagram that handle higher orders, such as the *k*-th order, where each



Voronoi Diagram, Figure 1 An example of the Voronoi Diagram [1]



Voronoi Diagram, Figure 2 A subsection of the Voronoi Diagram from Fig. 1. The shaded region represents all the points that are closest to *A* than *B*, *C*, and *D*. The point *X* denotes the midpoint between *C* and *D*, thus creating the hyperplane between the two. (Note: Modified diagram from [1]) point has *k* closest generating points. Another variant is the Farthest-Site, which defines where each point has the same farthest generated point. Several different distance metrics can be applied to these diagrams, the most popular of which are Euclidean and Manhatten distances.

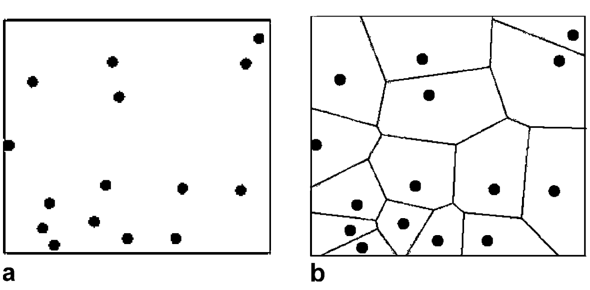
Historical Background

Voronoi diagrams were named after the mathematician Georgy Fedoseevich Voronoi of the Ukraine in 1905. However, these types of diagrams were used prior to the coined name of "Voronoi" in 1644 by Rene Descartes. They were also used in 1850 by a German mathematician, Johann Peter Gustav Lejeune Dirchlet, when he applied them to two- and three-dimensional data sets [2].

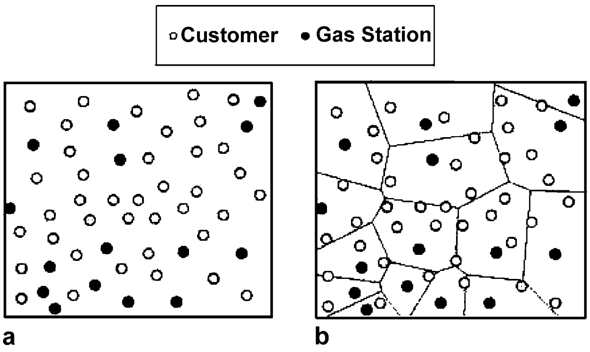
Scientific Fundamentals

Figure 3 shows a simple example of how a Voronoi Diagram is created. Figure 3a contains the set of generated objects in some spatial space. The first step is to find all the points that are the nearest to the generated object. Also, we need to find those points that may have more than one closest generated point (these points will lie on the perimeter of each polygon). Once these objects are known, we can then find the polygonal shapes around each object shown in Fig. 3b. Notice that any point within a polygonal shape is closer to the generated point inside the polygon than any other generated point in the data set. Thus, the polygonal shapes in Fig. 3b comprise our completed first order Voronoi Diagram.

Suppose we were the owner of several gas stations across Minneapolis and we had the capability of sending text messages to nearby customer's cell phones. Figure 4a shows this scenario where the black dots represent the gas stations and the white dots are the customers. Also, suppose we want to save money and limit the amount of text messages we send out from each gas station to only those customers that are closest to a certain gas station and not to any other gas station. Also, we want to issue a "snapshot" query for a certain time stamp. A possible solution to this problem is to create a Voronoi Diagram over Fig. 4a to produce Fig. 4b. This would show us which customers we can send advertisements to for the given time instance and be efficient in sending text messages to the best located customers. By contrast, if the customers were constantly moving while constantly sending text messages would create a continuos type of a problem. This would be a harder problem to solve and the use of Voronoi Diagrams as a "snapshot" would cause redundant calculations. The solution to this more difficult problem is discussed in Key Applications.



Voronoi Diagram, Figure 3 An example of the Voronoi Diagram creation [1]



Voronoi Diagram, Figure 4 A practical example of using Voronoi Diagrams. (Note: Modified diagram from [1])

Key Applications

Voronoi diagrams are used in many application domains, most notably the sciences and domains using spatial databases such as geography.

Sciences

The first domain area to use Voronoi Diagrams is the Sciences. The science domains discussed here are astronomy, biology, chemistry, forestry, geology, and medicine.

Astronomy Voronoi diagrams can be applied to astronomy when finding the center stars within galaxies and the area of each galaxy. With this information, galaxies can be computed and contrasted with each other using Voronoi diagrams [3]. Clusters of galaxies can be found using Voronoi tessellations by observing the galaxy's local density and its neighboring galaxies [2].

Biology Biologists have represented different species of plants using Voronoi Diagrams by characterizing them by their size ratio and location. They have shown that Voronoi diagrams can best represent the change in these values dur-

V

ing plant growth [4]. Matallic catalysts have been observed from an X-ray using Voronoi diagrams [5].

Forestry Plant dynamics are models over polygons or through Voronoi Tessellations (synonym to Voronoi Diagrams) in the populations found in French Guiana [6]. Also, plans for handling forest fires and locating people have utilized Voronoi diagrams by using the geometric nature in Green Islands [7].

Geology Geologists have simulated a volcano's lava flow by partitioning the surface into several Voronoi polygons [8]. Also, the prediction of reservoirs under several operational instances can be determined using Voronoi cells [9]. Voronoi cells are simply defined by each enclosed region within a Voronoi diagram.

Medicine The process of finding the best location for a hospital based on the locations of surrounding residents can be obtained by using weighted Voronoi diagrams [10]. A weighted Voronoi diagram is similar to the classical definition except that each cell has a weight and is added as part of the Euclidean distance from the points within the cell to the generated point. Also, the creation of casts for broken bones can be created by approximating the solidification patterns using Voronoi Diagrams [11].

Spatial Data

Voronoi diagrams are a very common partitioning technique used with spatial data. This section discusses spatial data applications within the domains of geography, geophysics, graph theory, the nearest neighbor problem, and route planning.

Geography Research in the domain of geography relies heavily on methods used in the spatial database domain. Voronoi diagrams can be used to create index structures for maps such as in [2]. Location Based Services are used in Terrain Modeling, where spatial regions are partitioned into polygons with each polygon representing a GPS type of a device [12].

Graph Theory In general, graphs are a set of nodes or objects connected by links or edges. Voronoi diagrams can be represented as graphs where the intersection of lines are the nodes and the perimeter connecting the intersections are the edges. Researchers in [13] proposed a method using graphs to ensure that each cell of a Voronoi diagram is connected. These diagrams can be shown to be efficiently computed using the principles in graph theory and can also be used by other methods such as the sweepline algorithm [2].

Nearest Neighbor Problem An interesting problem that lies in the spatial database domain is the Nearest Neighbor problem. Extensive study within this problem domain has led some to look to Voronoi diagrams and how they could be applied. The nearest neighbor problem can be defined as for a set of objects o and query point q in a spatial region, find all objects that are the nearest to q. This problem can be extended to the k-Nearest Neighbor (k-NN), where k is the k number of objects that are closest to q. Examples of k-NN application include finding the closest gas station to your location (1-NN) or the three closest restaurants based on your position (3-NN). Other extensions include the Reverse Nearest Neighbor and Group Nearest Neighbor.

In [14], Voronoi Diagrams were applied to the problem of finding the Nearest Neighbor by partitioning the data set into several regions. Based on these regions, the distances are precomputed within the regions and across the regions. Thus, when the nearest neighbors are determined, these distances are used to approximate which objects are the closest to a query object. An approximate approach is defined by estimating the location of an object based on some pre-defined region (i. e., Voronoi Diagram) or the trajectory of an object. This work assumes that the objects are moving and, therefore, that the queries have to be monitored continuously. This approach resolved the difficult problem mentioned in the Scientific Fundamentals section.

Route Planning In general, route planning is similar to the various applications found on the internet that find directions from one location to another, such as finding the quickest path within a transportation network using Voronoi diagrams [2]. Researchers in [15] developed a robot to freely roam in a one level building. To account for all the obstacles on the floor, Voronoi Diagrams were used to partition the floor to identify each object and its location. These objects or obstructions could be completely avoided by using an incremental approach to Voronoi diagrams [2].

Future Directions

In general, anything that can be represented spatially, can use some form of Voronoi diagrams. Marketing researchers can use them to create geographical representations of sales. Zoologists can use them to identify habitats of various species. In short, the list of potential applications is endless.

Voronoi Diagrams are especially important to the spatial database domain since they are a fundamental concept in partitioning a large spatial data set. Even though the concept is over 400 years old, new and exciting applications and variations are being developed everyday.

Cross References

► Distance Metrics
► Voronoi Diagrams for Query Processing
► Voronoi Terminology

Recommended Reading

1. R Development Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2004) 3-900051-07-0
2. Aurenhammer, F.: Voronoi diagrams survey of a fundamental geometric data structure. In: ACM Computing Surveys, vol. 23, pp. 345–405. (1991)
3. Zaninetti, L.: The galaxies distribution as given from the voronoi diagrams. In: Journal of Astronomy and Astrophysics, Italy
4. Gavrilova, M.: Weighted voronoi diagrams in biology. http://pages.cpsc.ucalgary.ca/~marina/vpplants/
5. Brumberger, H., Goodisma, J.: Voronoi cells: An interesting and potentially useful cell model for interpreting the small-angle scattering of catalysts. In: Journal of Applied Crystallography, vol. 16, pp. 83–88. (1983)
6. Mercier, F., Baujard, O.: Voronoi diagrams to model forest dynamics in french guiana. In: 2rd International Conference on GeoComputation University of Otago, (1997)
7. Roque, W.L., Choset, H.: The green island formation in forest fire modeling with voronoi diagrams. In: the 3rd Center for Geometric Computing Workshop on Compuational Geometry, (1998)
8. Stora, D., Agliati, P.O., Cani, M.P., Neyret, F., Gascuel, J.D.: Animating lava flows. In: Graphics Interface, (1999)
9. Christie, M., Subbey, S., Sambridge, M.: Prediction under uncertainty in reservoir modeling. In: the European Conference on the Mathematics of Oil Recovery, (2002)
10. Rezende, F.A.V.S., Almeida, R.M.V., Nobre, F.F.: Defining catchment areas for public hospitals in the municipality of rio de janeiro through weighted voronoi diagrams. In Cadernos de Saude Publica, vol. 16, pp. 467–475. (2000)
11. Houser, S.A.: Medical Surface Transformations for Rapid Approximation of Casting Solidification. MS in mechanical engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia (1996)
12. McGeough, J.: Location-based services and topology. In: GITA (2001)
13. Klein R., Wood, D.: Voronoi diagrams based on general metrics in the plane. In: Proceedings of the 5th Annual Symposium on Theoretical Aspects of Computer Science, pp. 281–291 (1998)
14. Kolahdouzan M., Shahabi, C.: Voronoi-based k nearest neighbor search for spatial network databases. In: VLDB (2004)
15. Blaer, P.: Robot path planning using generalized voronoi diagrams. http://www.cs.columbia.edu/~pblaer/projects/path_planner/

Voronoi Diagrams for Query Processing

CYRUS SHAHABI, MEHDI SHARIFZADEH
Information Laboratory (InfoLab) Computer Science Department, University of Southern California, Los Angeles, CA, USA

Synonyms

Voronoi-based query processing in road network databases; Network voronoi

Definition

Query processing in Spatial Network Databases (SNDBs) involves the following challenges: First, the distances between objects depend on their network connectivity and it is computationally expensive to compute these distances (e. g., shortest paths). Second, even when the distance can be approximated by simple vector distances, the query itself involves complex distance-based optimization problems. Several studies propose the use of Voronoi diagrams for spatial query processing in spatial networks. The ability of Voronoi diagrams to encode different geometric relationships based on arbitrary distance metrics in various spaces makes them ideal utilities to address the above challenges.

This entry shows the effectiveness of *network Voronoi diagrams* in answering two different spatial queries as case studies: k Nearest Neighbor (kNN) query finds the k closest spatial objects to a given query object, and Optimal Sequenced Route (OSR) query strives to find a route of minimum length starting from a given source location and passing through a number of *typed* locations in a specific sequence imposed on the types of the locations. Novel approaches to efficiently and accurately evaluate kNN and OSR queries in spatial network databases using variations of network Voronoi diagrams have been studied. The experimental results verify that these Voronoi-based query processing approaches outperforms their competitor index-based approaches in terms of query response time.

Historical Background

Although Voronoi diagrams were first used in 1644 by Rene Descartes, they were named after the mathematician Georgy Fedoseevich Voronoi in 1905. In 1850, a German mathematician Johann Peter Gustav Lejeune Dirchlet applied them to 2- and 3-dimensional points [7]. Since the introduction of Voronoi diagrams, different variations of these diagrams have been theoretically studied in the field of Computational Geometry. Many research works in the database community have also employed the underlying concept or the actual variations of Voronoi diagrams to address spatial queries in both vector spaces and the metric space pertaining to spatial network databases [3,4,5,11,12,13,15].

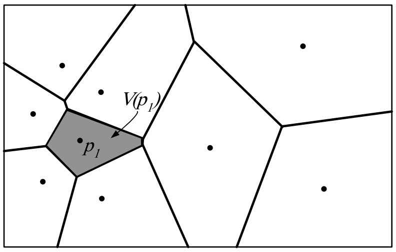
Scientific Fundamentals

With spatial network databases (SNDBs), spatial objects are restricted to pre-defined locations (e. g., points within the roads of a road network) that are specified by an underlying network. Consequently, the distance between each two locations is their network distance (e. g., the length of the shortest path through network edges that connects the two locations). This computationally expensive *network distance metric* depends on the connectivity of the network rather than the objects' locations.

Similar to general spatial databases, an SNDB carries geometric representations and hence demands special data processing techniques. Spatial queries seek for data objects possessing a specific *geometric relation* with a given spatial object or among themselves. This relation is mainly specified via a query function in terms of the *distance metric* defined in the space of data. As an example, consider a Nearest Neighbor (NN) query that returns the closest restaurant to a given location in Los Angeles. This query is formalized as finding the data object that *minimizes the distance* to a given object. That is, it exposes the *closeness* relation among the data objects with respect to a given object considering a certain distance metric.

Database research community has proposed spatial index structures such as R-trees [1] for efficient processing of spatial queries. These index structures respect the geometric relationship of the data objects in order to organize them into groups of distance-related objects. However, they are limited to specific distance metrics in vector spaces (e. g., Euclidean in $\mathbb{R}^2$) and hence cannot represent arbitrary distance metrics (e. g., network distance) in general metric spaces. Consequently, index-based query processing in SNDBs requires additional computation on the relevant data retrieved after traversing the index; the index is only utilized to prune large portions of data which are not likely to be in the query result.

The fact that almost all spatial queries discover distance-based functions motivates the use of a data structure customizable for spatial queries in spaces such as spatial networks. A data structure that nicely captures varieties of distance-based relations in a geometric space is Voronoi diagram [7]. Given a set of spatial data objects, a general Voronoi diagram uniquely partitions the space of objects into disjoint regions where the closest data object of all the objects inside a single region is the same. That is, the region corresponding to an object p covers the objects in space that are closer to p than to any other data object. Figure 1 shows the *ordinary* Voronoi diagram of nine points in $\mathbb{R}^2$ where the distance metric is Euclidean. It is clearly seen how the concept of closeness (according to Euclidean distance) exploited in the query for Los Angeles restaurants is geometrically encoded in this diagram if one consider the seven depicted points as restaurant locations. If our given query location is inside the polygon containing p_i then its closest restaurant is the restaurant located at p_i. That is, ordinary Voronoi diagrams can answer NN queries utilizing a straightforward efficient function that examines the inclusion of a point inside a polygon.



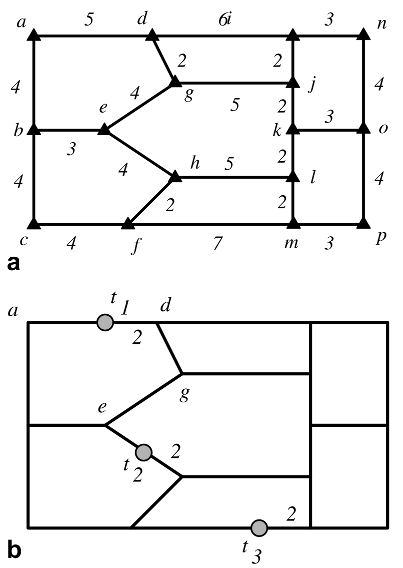
Voronoi Diagrams for Query Processing, Figure 1 Example of an ordinary Voronoi diagram with respect to Euclidean distance

Different variations of Voronoi diagrams have been theoretically studied in the field of Computational Geometry. Each of these variations captures a specific relation utilizing a different distance metric in a particular space. Identifying the corresponding distance-based relation, many spatial queries can be easily mapped to the corresponding Voronoi diagram. This entry uses variations of *network* Voronoi diagrams for spatial query processing in the space of SNDBs. To show the flexibility of Voronoi diagrams with respect to their two parameters of *space* and *distance*, the entry studies two totally different spatial queries as its case studies: k Nearest Neighbor (kNN) query and Optimal Sequenced Route (OSR) query.

Modeling Spatial Networks

This section formally models the space of spatial networks. Consider the weighted undirected[1] graph $G = (V, E)$ as the two sets V of vertices, and $E \subseteq V \times V$ of edges. Each edge of E, directly connecting vertices u and v, is represented as the pair $[u,v]$. Each vertex v represents a 2-d point $(v.x, v.y)$ in a geometric space (e. g., an intersection in a road network). Hence, each edge is also a line segment in that space (e. g., a road segment). A numeric

[1]Many spatial networks consist of directed edges and hence must be modeled as directed graphs. This entry uses undirected graphs for simplicity. However, the concepts can be easily revisited to support directed graphs.



Voronoi Diagrams for Query Processing, Figure 2 **a** Graph model of a road network, **b** 3 points on the edges of the graph

weight (cost) w_{uv} is associated with the edge $[u,v]$. In an SNDB representing a road network, this is the distance or the travel time between intersections u and v. The set N refers to the space of points located on the edges/vertices of graph G. For a point $p \in N$ located on the edge $[u,v]$, define $w_{up} = \frac{|up|}{|uv|} w_{uv}$ where $|uv|$ is the Euclidean distance between u and v. Figure 2a shows the graph model of a road network including the vertex set $V = \{a, \ldots, p\}$. Each edge of the graph is labeled by its weight. Figure 2b shows points $t_1, \ldots, t_3$ on the edges of the same graph. As shown in the figure, point $t_1 \in N$ corresponds to the weights $w_{t_1 a} = 3$ and $w_{t_1 d} = 2$.

Definition 1 Given a graph G, a *path* P from $p_1 \in N$ to $p_2 \in N$ is an ordered set $P = \{p_1, v_1, \ldots, v_n, p_2\}$ consisting of a sequence of connected edges from p_1 to p_2. Here, p_1 and p_2 are located on the edges $[u,v_1]$ and $[v_n, w]$, respectively. Also, v_i is connected to v_{i+1} by the edge $[v_i, v_{i+1}]$ for $1 \leq i < n$. As shown in Fig. 2b, $P = \{t_1, d, g, e, t_2\}$ is a path from t_1 to t_2.

Definition 2 Given a path $P = \{p_1, v_1, \ldots, v_n, p_2\}$, define *Path Cost* of P, *pcost(P)*, as the sum of the costs of all edges in P. Formally, for the path P,

$$pcost(P) = w_{p_1 v_1} + \sum_{i=1}^{n-1} w_{v_i v_{i+1}} + w_{v_n p_2} .$$

In Fig. 2b, the cost of path $P = \{t_1, d, g, e, t_2\}$ is calculated as $pcost(P) = 2 + 2 + 4 + 2 = 10$. For the points p_1 and $p_2 \in N$, $P_{p_1 p_2}$ denotes the *shortest path* from p_1 to p_2 in G; the path $P = \{p_1, \ldots, p_2\}$ with minimum cost *pcost(P)*.

Definition 3 Given the two points p_1 and p_2 in N, the *network distance* between p_1 and p_2, $D_n(p_1, p_2)$, is the cost of the shortest path between p_1 and p_2 (i. e., $D_n(p_1, p_2) = pcost(P_{p_1 p_2})$). For instance, one have $D_n(t_1, t_2) = 10$. The network distance defined by $D_n(.,.)$ obeys the triangular inequality.

Network Voronoi Diagrams

Given a graph G, let $S = \{p_1, \ldots, p_n\}$ be a set of points in N each assigned a numeric weight $w(p_i)$. The *Additively Weighted (AW-) network Voronoi diagram* of S partitions N into regions. Each region corresponding to a point $p \in S$ is called the *network Voronoi cell* of p and includes edges or portions of edges of G. Each cell *V(p)* includes all points in N with a common closest point in the given set S according to a distance metric. It contains all the points of N for which have

$$\forall p\prime \in S, p\prime \neq p \Rightarrow D_n(q, p) + w(p) \leq D_n(q, p\prime) + w(p\prime). \quad (1)$$

The set given by

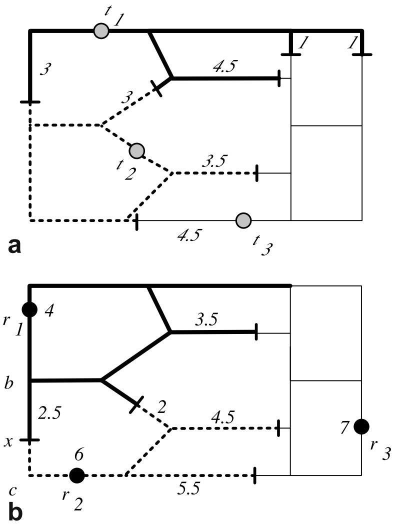
$$VD(P) = \{V(p_1), \ldots, V(p_n)\} \quad (2)$$

is called the AW-network Voronoi diagram generated by S with respect to the distance function $D_n(.,.)$. When all the weights *w(p)* are set to zero, the diagram is *ordinary* network Voronoi diagram of S [7]. Figure 3a shows the ordinary network Voronoi diagram of the points $t_1, \ldots, t_3$. The network Voronoi cell of each t_i is depicted using a different line style. The figure shows separator markers to specify the boundaries of adjacent cells. Figure 3b also illustrates the AW-network Voronoi diagram of the points r_1, r_2, r_3 each labeled by its weight.

Spatial Query Case Studies

This section discusses the use of network Voronoi diagrams in processing two spatial queries in SNDBs.

***k* Nearest Neighbor Query** The first case study uses ordinary network Voronoi diagrams to process k Nearest Neighbor (kNN) queries. Many researchers have focused on the problem of kNN queries in spatial databases. This type of query is frequently used in Geographical Information Systems and is defined as: given a set of spatial objects (or points of interest), and a query point, find the k closest objects to the query. An example of kNN query is a query initiated by a GPS device in a vehicle to find the 5 closest restaurants to the vehicle. With SNDBs, the realistic



Voronoi Diagrams for Query Processing, Figure 3 AW-network Voronoi diagrams of **a** $\{t_1, t_2, t_3\}$, **b** $\{r_1, r_2, r_3\}$

distance between objects (e. g., the vehicle and the restaurants) is their actual network distance.

The application of the Voronoi diagrams to *k*NN queries have been extensively studied in Computational Geometry. The solution is based on calculating the order-*k* Voronoi diagrams of a network. This solution is impractical for real-world scenarios since it requires that the value of *k* be predetermined. Moreover, the order-*k* Voronoi cells have complex shapes and the corresponding algorithms have a very high complexity. In [4], Shahabi and Kolahdouzan proposed an approach, termed Voronoi-based Network Nearest Neighbor (VN^3), that utilizes the ordinary network Voronoi diagrams to answer the *first* nearest neighbors of a query point. With their approach, the other $(k-1)$ neighbors are found efficiently by utilizing a set of proven properties and the pre-computed distances.

The main idea behind VN^3 is to first partition a large network in to smaller/more manageable regions. VN^3 achieves this by generating an ordinary network Voronoi diagram over the points of interest (e. g., restaurants). Next, it pre-computes the intra and inter distances for each Voronoi cell. That is, for each cell, it pre-computes the distances between all the edges (i. e., *border* points on the boundary) of the cell to its center. VN^3 also pre-computes distances only across the border points of the adjacent cells. This will reduce the pre-computation time and space by localizing the computation to cells and handful of neighbor-cell node-pairs. Now, to find the *k* nearest-neighbors of a query object q, VN^3 first finds the first nearest neighbor by simply locating the Voronoi cell that contains q. This can be easily achieved by utilizing a spatial index (e. g., R-tree) that is generated for the Voronoi cells. In [4], they prove that the next nearest neighbors of q are within the adjacent cells of the previously explored ones, which can be efficiently retrieved from a lookup table. VN^3 then utilizes the intra-cell pre-computed distances to find the distance from q to the borders of the Voronoi cell of each candidate, and finally the inter-cell pre-computed distances to compute the actual network distance from q to each candidate. The local pre-computation nature of VN^3 also results in low complexity of updates when the network is modified.

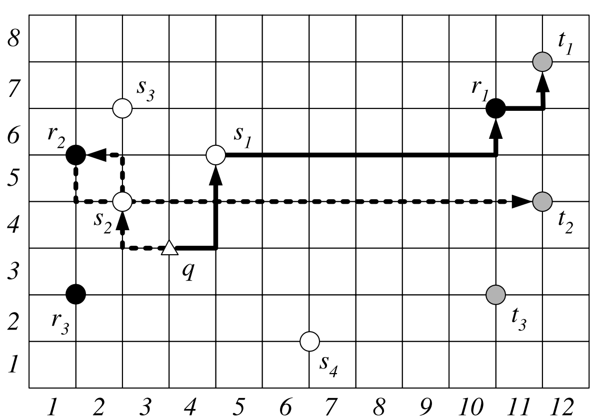
The empirical experiments with real-world datasets shows that VN^3 outperforms the Incremental Network Expansion (INE) approach [8] in query processing time by a factor of *1.5* to *12* depending on the density of the points of interest. VN^3's performance is shown to be independent of the density and distribution of the points of interest, and the location of the query object [4]. It is also shown in [4] that the required computation and space for the pre-computation component of VN^3 is three orders of magnitude less than that of the naive solution that pre-computes all the node-pair distances.

Optimal Sequenced Route Query As a representative of the spatial queries defined based on complex distance functions, the second case study, shows utilizing AW-network Voronoi diagrams to address the Optimal Sequenced Route (OSR) queries. Sharifzadeh et al. introduced OSR query in [9] as follows. Suppose that one is planning a Saturday trip in town as following: first he/she intends to visit a shopping center in the afternoon to check the season's new arrivals, then he/she plans to dine in an Italian restaurant in early evening, and finally, he/she would like to watch a specific movie at late night. Naturally, the user intends to drive the minimum overall distance to these destinations. In other words, he/she needs to find the locations of the shopping center s_i, the Italian restaurant r_j, and the theater t_k that shows the specific movie, which driving toward them considering the sequence of the plan shortens his/her trip (in terms of distance or time). Note that in this example, a time constraint enforces the order in which these destinations are to be visited; the user usually does not have dinner in the afternoon, or does not go for shopping at late night. Figure 4 shows three different types of *point sets* shown by white, black and gray circles, which represent shopping centers, Italian restaurants, and theaters, respectively, and a starting point q (shown by $\triangle$). The OSR from q to a *white* s_i, a *black* r_i, and finally a *gray* t_i point is (s_1, r_1, t_1).

With OSR query, the order of the sequence in which some points must be visited is *enforced* and cannot be changed. The query is essential in other application domains such as *crisis management*, *air traffic flow management*, *supply chain management*, and *video surveillance*. Planning queries similar to OSR have received recent attention by the database community [2,6,14] to solve problems in the domains of air traffic flow and supply chain management, which shows the importance of these query types.

In [11], Sharifzadeh and Shahabi show that unique regions of the space could be identified within which the result of a given OSR query with a certain sequence is invariant. Subsequently, they identify an additively weighted function as the distance function used by AW-Voronoi diagrams in order to encode these regions. They theoretically prove the relation between OSR queries and this variation of Voronoi diagrams and provide a novel query processing technique. In [10], they extend the approach to utilize AW-network Voronoi diagrams for the space of SNDBs.

Assume that one is only interested in the optimal sequenced routes that follow a given sequence M. For example, assume that this fixed sequence for Fig. 4 is *(white, black, gray)*. Suppose he/she can partition the space of points of interest (e. g., N for graph model shown) into regions each including all the points with a common OSR. For instance, in Fig. 4, there are many points similar to q for which the optimal sequenced route to a *white*, then a *black*, and finally a *gray* point is the route (s_1, r_1, t_1). Suppose that the partitioning identifies the region including these points. Furthermore, assume that the partitioning also identifies the region corresponding to each and every possible route (s_i, r_j, t_k) to a *white*, a *black*, and a *gray* point. Assume that one associates and stores all regions and their corresponding OSRs. Therefore, for a given starting point q he/she can address the OSR query with sequence M by simply locating the *unique* region that includes q and reporting its corresponding optimal sequenced route. In [11], they prove that such partitioning exists as an *AW-Voronoi diagram* and theoretically exploits interesting properties of these diagrams which makes them appropriate data structures for efficient OSR query processing. [10] shows that the same approach is applicable for the space of SNDBs if it uses AW-network Voronoi diagrams.



Voronoi Diagrams for Query Processing, Figure 4 OSR from q to a *white*, a *black*, and then a *gray* point

In [11], the authors enhance the above query processing approach by exploiting the following interesting property of OSRs: if (s_1, r_1, t_1) is q's optimal route to a *white*, a *black*, and a *gray* point, then (r_1, t_1) is s_1's optimal route to a *black* and then a *gray* point. *Recursively*, (t_1) is r_1's optimal route to a *gray* point. This property enables the approach to avoid storing the complete OSR corresponding to each region of the diagram for a given sequence M; it stores only the first point of the OSR for each region of the AW-Voronoi diagram (e. g., s_1 for the region corresponding to (s_1, r_1, t_1)). Instead, the approach requires to compute and store the AW-Voronoi diagrams corresponding to all suffixes of M.

For instance, to answer queries with sequence *(white, black, gray*) in Fig. 4, the approach requires the diagrams corresponding to three sequences *(white, black, gray)*, *(black, gray)*, and *(gray*). These diagrams are referred to as the *OSR-Voronoi family* of sequence *(white, black, gray)* [11]. Now, the approach iteratively processes a given OSR query. For the starting point q, it first find s_1, the white point associated with the region including q in the diagram of *(white, black, gray)*. Then, it finds r_1, the black point stored with the region including s_1 in the diagram of *(black, gray)*. Finally, it seeks for t_1, the gray point corresponding to the region including r_1 in the diagram of *(gray)*. Now, the final OSR of q is the route (s_1, r_1, t_1), which includes s_1, r_1, and t_1 in the order of their retrieval.

The above OSR query processing using Voronoi diagrams best suits the applications where the sequences of user's interest are known in advance. This allows pre-computation of the corresponding diagrams. Notice that using the OSR-Voronoi family of a sequence M, one can address OSR queries of *all suffix sequences* of M. Through extensive experiments with a real-world dataset, [11] shows that its proposed approach significantly outperforms the R-LORD approach proposed in [9] in terms of response time. It also shows that the off-line process used to build the OSR-Voronoi family of a single sequence is reasonably fast.

Key Applications

Online Maps

Both kNN and OSR queries on SNDBs corresponding to real-world transportation networks are of great interest to the users of online maps. Many commercial online services such as Yahoo! Maps, MapQuest, Google Maps, and Microsoft Live Local provide very limited versions of *k*NN queries.

Trip Planning

A trip planning application usually provides facilities to schedule optimal routes of a trip. Different variations of *k*NN and OSR queries are essential modules for such an application that can be utilized for automatic plan generation.

Supply Chain Management

Optimal assignment of delivery packages to a fleet of trucks can benefit from the result of *k*NN queries issued on the dataset of delivery locations. Similarly, issuing an OSR query generates optimal delivery routes of desired sequence to typed locations for each of the trucks.

Crisis Management

Spatial queries on SNDBs play a prominent role in this application domain. As a scenario utilizing OSR queries, suppose that an ambulance needs to repeatedly visit one of the several attacked points a_i and hospitals h_j, respectively. The ambulance should visit as many of the attacked points as possible in the shortest time. The constraint that enforces the order in this example is that there is no reason for the ambulance to go to a hospital if it has not yet picked up an injured person. Note that in this example, although there are only two different types of points (i. e., attacked points and hospitals), the size of the sequence can become arbitrary large (e. g., $(a_1, h_1, a_2, h_2, \ldots, a_i, h_i)$).

Future Directions

Voronoi diagrams can be used to answer more complex types of spatial queries in the metric space of SNDBs. Group and continuous nearest neighbor query types are potential candidates. Another promising direction for the future work is to investigate using AW-network Voronoi diagrams for other variations of trip planning queries. An example of these queries is a variation of OSR query where the sequence is not important (TPQ problem [6]). An orthogonal problem is studying efficient algorithms to store Voronoi diagrams in secondary storage.

Cross References

► Nearest Neighbor Queries in Network Databases
► Road Network Data Model
► Spatio-temporal Queries on Road Networks, Coding Based Methods
► Trip Planning Queries in Road Network Databases
► Voronoi Diagram

Recommended Reading

1. Guttman, A.: R-trees: a dynamic index structure for spatial searching. In: SIGMOD '84: Proceedings of the 1984 ACM SIGMOD international conference on Management of data, New York, NY, USA, pp. 47–57. ACM Press (1984)
2. Hadjieleftheriou, M., Kollios, G., Bakalov, P., Tsotras, V.J.: Complex Spatio-temporal Pattern Queries. In: Proceedings of the 31st International Conference on Very Large Data Bases (VLDB'05), pp. 877–888 (2005)
3. Hagedoorn, M.: Nearest Neighbors Can Be Found Efficiently If the Dimension Is Small Relative to the Input Size. In: Proceedings of the 9th International Conference on Database Theory - ICDT 2003. volume 2572 of Lecture Notes in Computer Science, pp. 440–454, Springer, January (2003)
4. Kolahdouzan, M., Shahabi, C.: Voronoi-Based K Nearest Neighbor Search for Spatial Network Databases. In: VLDB 2004, Toronto, Canada
5. Korn, F., Muthukrishnan, S.: Influence Sets Based on Reverse Nearest Neighbor Queries. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of data. pp. 201–212, ACM Press (2000)
6. Li, F., Cheng, D., Hadjieleftheriou, M., Kollios, G., Teng, S.H.: On Trip Planning Queries in Spatial Databases. In: Proceedings of the 9th International Symposium on Spatial and Temporal Databases (SSTD'05), volume 3633 of Lecture Notes in Computer Science, pages 273–290. Springer, August (2005)
7. Okabe, A., Boots, B., Sugihara, K., Chiu, S.N.: Spatial Tessellations, Concepts and Applications of Voronoi Diagrams, 2nd edn. John Wiley and Sons Ltd., (2000)
8. Papadias, D., Zhang, J., Mamoulis, N., Tao, Y.: Query Processing in Spatial Network Databases. In: VLDB 2003, Berlin, Germany
9. Sharifzadeh, M., Kolahdouzan, M., Shahabi, C.: The Optimal Sequenced Route Query. The VLDB Journal (2007)
10. Sharifzadeh, M., Shahabi, C.: Processing Optimal Sequenced Route Queries using Voronoi Diagrams. Submitted for review
11. Sharifzadeh, M., Shahabi, C.: Additively Weighted Voronoi Diagrams for Optimal Sequenced Route Queries. In: Proceedings of the 3rd International Workshop on Spatio-temporal Database Management (STDBM'06), September (2006). CEUR Workshop Proceedings
12. Sharifzadeh, M., Shahabi, C.: The Spatial Skyline Queries. In: Proceedings of the 32nd International Conference on Very Large Data Bases: VLDB'06. pp. 751–762 September (2006)
13. Stanoi, I., Riedewald, M., Agrawal, D., Abbadi, A.E.: Discovery of Influence Sets in Frequently Updated Databases. In: Proceedings of the 27th International Conference on Very Large Data Bases (VLDB'01) pp. 99–108. Morgan Kaufmann Publishers Inc. (2001)
14. Terrovitis, M., Bakiras, S., Papadias, D., Mouratidis, K.: Constrained Shortest Path Computation. In: Proceedings of the 9th International Symposium on Spatial and Temporal Databases (SSTD'05), volume 3633 of Lecture Notes in Computer Science, pages 181–199. Springer, August (2005)
15. Zhang, J., Zhu, M., Papadias, D., Tao, Y., Lee, D.L.: Location-Based Spatial Queries. In: Proceedings of the 2003 ACM SIGMOD international conference on Management of data. pp. 443–454. ACM Press (2003)

Voronoi Terminology

JAMES M. KANG
Department of Computer Science and Engineering,
University of Minnesota, Minneapolis, MN, USA

Definition

The Voronoi diagram contains a set of enclosed regions, each of which contains a single generated point. Each region can also be called a Voronoi cell or a Dirichlet domain. Every point other than the generated point in the Voronoi cell is closer to its respective generated point than any other Voronoi cell's generated point. For example, Fig. 2 contains four distinct Voronoi cells.

Main Text

Voronoi diagrams can be represented using multiple dimensions such as in a Voronoi-Dirichlet Polyhedron (VDP), a three-dimensional polytope containing a single point which is bounded by a set of planes. The same rules apply as in a two-dimensional Voronoi diagram where every point within the enclosed region is closer to the generated point than any other generated points in the overall spatial region. For example, a galaxy cluster has a three-dimensional shape and the center of the cluster could be the generated point. Thus, all the stars in this galaxy are closer to this center star than any other galaxy's center.

The Voronoi diagram contains a set of Voronoi cells. This set of Voronoi cells can also be referred to as a Voronoi tesselation. A tesselation is the arrangement of enclosed regions placed together in some form. An example could be a tiled floor where each tile is a Voronoi cell and the whole floor contains a Voronoi tesselation. An evenly tiled floor is considered as a regular tesselation where each cell is of the same shape. An un-evenly tiled floor where each tile is not symmetric with respect to other tiles is called an irregular tesselation.

Cross References

► Distance Metrics
► Voronoi Diagram
► Voronoi Diagrams for Query Processing

Voronoi Tesselation

► Voronoi Diagram

Voronoi-Based Query Processing in Road Network Databases

► Voronoi Diagrams for Query Processing

Voxel

► Spatial Data Transfer Standard (SDTS)

Walking, Joint

► Participatory Planning and GIS

Wavelets

► Biomedical Data Mining, Spatial

Wayfinding: Affordances and Agent Simulation

MARTIN RAUBAL
Department of Geography, University of California-Santa Barbara, Santa Barbara, CA, USA

Synonyms

Wayfinding behavior; Affordance; Agent simulation; Environmental communication; Cognitive psychology; Multi agent systems

Definition

Wayfinding behavior is the purposeful, directed, and motivated movement from an origin to a specific distant destination that cannot be directly perceived by the traveler. It involves interaction between the wayfinder and the environment.

Affordances are a concept from ecological psychology based on the paradigm of direct perception. They are specific combinations of the properties of substances and surfaces taken with reference to an observer. These invariant compounds are specified in ambient light—which is the result of illumination—and detected as units. Ambient light has structure and therefore information.

Agent simulation is a technique of imitating the behavior of some situation or process involving one or many agents. An agent is anything that can perceive its environment through sensors and act upon it through effectors. Agents are situated in some environment and capable of autonomous action.

Historical Background

Kevin Lynch's [1] "The Image of the City" is the first documentation of *human wayfinding* research in the literature. His goal was to develop a method for the evaluation of city form based on the concept of *imageability*, and to offer principles for city design. As part of the interviews people had to perform mental trips across their cities, describing the sequence of things and landmarks they would see along the way. Based on his investigations Lynch divided the contents of the city images into five classes: paths, edges, districts, nodes, and landmarks. These elements were described as the building blocks in the process of making firm, differentiated structures at the urban scale and have been the basis for later wayfinding research.

The term *affordance* was originally introduced by James J. Gibson who investigated how people visually perceive their environment [2]. Affordances belong to the ecological approach to psychology, which was developed to solve the major problem of cognitive psychology—the problem of knowledge. It is based on ecological science, a multidisciplinary advance to the study of living systems, their environments, and the reciprocity between the two.

Agent simulation is a part of the larger area of computer simulation, which was developed in parallel with the rapid growth of computers starting with the Manhattan Project in the 1940s. There, the process of nuclear detonation was simulated using a Monte Carlo algorithm. The term *agent* has become popular in the area of Artificial Intelligence.

Scientific Fundamentals

Wayfinding

Research in human wayfinding investigates the processes that take place when people orient themselves and navigate through space. Theories explain how people find their ways in the physical world, what people need to find their ways, how they communicate directions, and how people's verbal and visual abilities influence wayfinding. Wayfinding behavior is described as purposeful, directed, and motivated movement from an origin to a specific distant destination that cannot be directly perceived by the traveler [3,4]. Such behavior involves interactions between the traveler and the environment, such as moving. Hence, wayfinding takes place in large-scale spaces [5]. Such spaces cannot be perceived from a single viewpoint; therefore, people have to navigate through large-scale spaces to experience them. Examples for large-scale spaces are landscapes, cities, and buildings.

People use various spatial and cognitive abilities to find their ways. These abilities are a necessary prerequisite to use environmental information or representations of spatial knowledge about the environment. The spatial abilities are task-dependent and seem to involve mainly four interactive resources: perceptual capabilities, information-processing capabilities, previously acquired knowledge, and motor capabilities [3]. As for the spatial abilities, the cognitive abilities also depend on the task at hand. Finding one's way in a city uses a different set of cognitive abilities than wayfinding in a building. Three categories of wayfinding tasks can be distinguished [3]: travel with the goal of reaching a familiar destination, exploratory travel with the goal of returning to a familiar point of origin, and travel with the goal of reaching a novel destination. A task within the last category is most often performed through the use of symbolic information.

The literature on human wayfinding performance discusses empirical results of how people find their ways. Investigations are based on collecting individuals' perceptions of distances, angles, or locations. Weisman [6] identified four classes of environmental variables that influence wayfinding performance in built environments: visual access; architectural differentiation; signs and room numbers to provide identification or directional information; and plan configuration. Seidel's [7] study at the Dallas/Fort Worth Airport showed that the spatial structure of the physical environment has a strong influence on people's wayfinding behavior. People's familiarity with the environment also has a big impact on wayfinding performance.

Research on people's wayfinding performance helped to establish practical guidelines on how to design public buildings to facilitate wayfinding. Arthur and Passini [8] introduced the term *environmental communication*, arguing that the built environment and its parts should function as a communication device. They mention two major aspects regarding the understanding of buildings: a *spatial* aspect that refers to the total dimensions of the building and a *sequential* aspect that considers a building in terms of its destination routes. Destination routes should eventually lead to destination zones. These are groupings of similar destinations within buildings into clearly identifiable zones. In order to facilitate wayfinding to such destination zones the circulation system should be of a form people can easily understand.

Affordances

The theory of affordances [2] is based on ecological psychology, which advocates that knowing is a direct process: The perceptual system extracts invariants embodying the ecologically significant properties of the perceiver's world. Gibson's theory is based on the tenet that animal and environment form an inseparable pair. This complementarity is implied by Gibson's use of ecological physics. Such physics considers functions of the environment at an ecological size level contrary to a description in terms of space, time, matter, etc., within classical physics. Affordances have to be described relative to the person. For example, a chair's affordance "to sit" results from a bundle of attributes, such as "flat and hard surface" and "height", many of which are relative to the size of an individual. Later work with affordances builds on this so-called agent-environment mutuality.

Affordances can be considered as measurable aspects of the environment, but only to be measured in relation to the individual. It is particularly important to understand the action relevant properties of the environment in terms of values intrinsic to the agent. Warren [9] demonstrates that the "climbability" affordance of stairs is more effectively specified as a ratio of riser height to leg length. Experimentally, subjects of different heights perceived stairs as climbable depending on their own leg length, as opposed to some extrinsically quantified value. Additionally, dynamic or task specific conditions must be considered.

Many researchers have believed that Gibson's theory is insufficient to explain perception because it neglects processes of cognition. His account deals only with individual phenomena, but ignores categories of phenomena [10]. Norman [11] investigated affordances of everyday things, such as doors, telephones, and radios, and argued that they provide strong clues to their operation. He recast affordances as the results from the mental interpretation of things, based on people's past knowledge and experiences, which are applied to the perception of these things. Gaver [12] stated that a person's culture, social setting, experience, and intentions also determine her perception of affordances. Affordances, therefore, play a key role in an experiential view of space, because they offer a user centered perspective. Similarly, Rasmussen and Pejtersen [13] pointed out that modeling the physical aspects of the environment provides only a part of the picture. "The framework must serve to represent both the physical work environment and the 'situational' interpretation of this environment by the actors involved, depending on their skills and values." [13, p. 122] This can be broken into three relevant parts, the mental strategies and capabilities of the agents, the tasks involved, and the material properties of the environment. In order to supplement Gibson's theory of perception with elements of cognition, situational aspects, and social constraints, Raubal [14] presented an extended theory of affordances suggesting that affordances belong to three different realms: physical, social-institutional, and mental. In a similar effort, the framework of distributed cognition was used to describe and explain the concept of affordance [15].

Agent Simulation for Wayfinding

Simulation of human behavior in space is a powerful research method to advance our understanding of the interaction between people and their environment. It allows for both the examination and testing of models and their underlying theory as well as the observation of the system's behavior [16].

According to the heterogeneity of the fields there is no common agreement about a definition of the term agent. An agent can be regarded as anything that perceives its environment through sensors and acts upon that environment through effectors [17]. Agents are situated in some environment and capable of autonomous action. Autonomy and the embedding into the environment are the two key properties of agents. Categories such as intelligent, distributed, and mobile agents exist.

Multi-agent systems (MAS) depict systems as a combination of multiple autonomous and independent agents and are therefore well suited to simulate the wayfinding behavior of different actors. Formally, the term multi-agent system refers to a system consisting of the following parts [18]:

- The *environment* E consisting of:
  - A set of *objects* O. Objects can be perceived, created, destroyed, and modified by agents.
  - A set of *agents* A. Agents are a subset of objects ($A \subseteq O$) capable of performing actions—the active entities of the system.
  - A set of *locations* L determining the possible positions of the objects (from the set O) in space.
- An assembly of *relations* R that link objects and agents.
- A set of *operations* Op enabling the possibility for agents to perceive, manipulate, create, and destroy objects of O, in particular representing the agents' actions.
- A set of *operators* U with the task of representing the application of the operations from Op and the reactions of the world to this attempt of modification. The operators from U are called the *laws of the universe*.

Agents have been mainly dealt with in Artificial Intelligence but have recently also gained popularity in other fields such as geography. MAS are of interest to simulating various human activities in the geo-domain due to their ability to reflect human behavior.

Key Applications

Agent-based simulation for human wayfinding has been used to simulate people's wayfinding behavior in spatial information and design systems. It can help to determine where and why people face wayfinding difficulties and what needs to be done to avoid them. Affordances have been implemented in such agent-based frameworks to model the agent's behavior in a cognitively plausible way [19]. For example, paths, which are clearly discernible—through markings on the ground or guiding structures on the side or above—facilitate visually controlled locomotion, which is directed by visual perception and depends on sequential optical information. In other words, paths afford moving along (doorways afford entering, columns afford obstructing, etc.). It has also been shown that integrating the affordance theory into agent architectures is an elegant solution to the problem of providing both rapid scenario development and the simulation of individual differences in perception, culture, and emotionality [20].

Agent simulation can also be employed for the analysis of geospatial problems related to wayfinding, such as the behavior of customers shopping in a mall [21], spatial communication with maps [22], and wayfinding in virtual spaces [23]. Other application domains for agent simulation of wayfinding include pedestrian traffic flow [24], and crowd and evacuation simulation [25].

Future Directions

In the future wayfinding research will focus more deeply on commonalities and differences between wayfinding in the real world compared to wayfinding in electronic and virtual spaces. Finding the particularities with regard to human spatial cognition will help designing more user-friendly wayfinding systems. The concept of affordances needs to be further developed to account for social and cognitive processes, and also with regard to its representation in computer systems. This way the concept will become more useful, both in geospatial system design and with regard to various aspects of Artificial Intelligence and robotics. Agent simulation of wayfinding will be extended to cover more and different application domains and therefore help in testing wayfinding models.

Recommended Reading

1. Lynch, K.: The Image of the City. MIT Press, Cambridge, Massachusetts (1960)
2. Gibson, J.: The Ecological Approach to Visual Perception. 332. Houghton Mifflin Company, Boston (1979)
3. Allen, G.: Spatial Abilities, Cognitive Maps, and Wayfinding – Bases for Individual Differences in Spatial Cognition and Behavior. In: Golledge, R. (ed.) Wayfinding Behavior – Cognitive Mapping and Other Spatial Processes, p. 46–80. Johns Hopkins University Press, Baltimore (1999)
4. Golledge, R.: Human Wayfinding and Cognitive Maps. In: Golledge, R. (ed.) Wayfinding Behavior – Cognitive Mapping and Other Spatial Processes, p. 5–45. Johns Hopkins University Press, Baltimore (1999)
5. Montello, D.: Scale and Multiple Psychologies of Space. In: Frank, A.U., Campari, I. (eds.) Spatial Information Theory: Theoretical Basis for GIS, p. 312–321. Springer-Verlag, Heidelberg-Berlin (1993)
6. Weisman, J.: Evaluating architectural legibility: Way-finding in the built environment. Environ. Behav. **13**, 189–204 (1981)
7. Seidel, A.: Way-Finding in Public Spaces: The Dallas/Fort Worth, USA Airport. 20th International Congress of Applied Psychology. Edinburgh, Scotland (1982)
8. Arthur, P., Passini, R.: Wayfinding: People, Signs, and Architecture, 238. McGraw-Hill Ryerson, Toronto (1992)
9. Warren, W.: Constructing an Econiche. In: Flack, J., et al. (eds.) Global Perspectives on the Ecology of Human-Machine Systems, p. 210–237. Lawrence Erlbaum Associates, Hillsdale, New Jersey (1995)
10. Lakoff, G.: Women, Fire, and Dangerous Things: What Categories Reveal About the Mind, 614. The University of Chicago Press, Chicago (1987)
11. Norman, D.: The Design of Everyday Things, 257. Doubleday, New York (1988)
12. Gaver, W.: Technology Affordances. Human Factors in Computing Systems, CHI'91 Conference Proceedings, p. 79–84. ACM Press, New York (1991)
13. Rasmussen, J., Pejtersen, A.: Virtual Ecology of Work. In: Flack, J., et al. (eds.) Global Perspectives on the Ecology of Human-Machine Systems, p. 121–156. Lawrence Erlbaum Associates, Hillsdale, New Jersey (1995)
14. Raubal, M.: Ontology and epistemology for agent-based wayfinding simulation. Int. J. Geogr. Inf. Sci. **15**(7), 653–665 (2001)
15. Zhang, J., Patel, V.: Distributed cognition, representation, and affordance. Pragmatics & Cognition **14**(2), 333–341 (2006)
16. Gimblett, H., Itami, R., Durnota, D.: Some Practical Issues in Designing and Calibrating Artificial Human-Recreator Agents in GIS-based Simulated Worlds. Complexity International, 3 (1996)
17. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. 2nd edn. Prentice Hall Series in Artificial Intelligence. Prentice Hall, London (2003)
18. Ferber, J.: Multi-Agent Systems – An Introduction to Distributed Artificial Intelligence. Addison-Wesley, London, UK (1999)
19. Raubal, M.: Human wayfinding in unfamiliar buildings: a simulation with a cognizing agent. Cognitive Processing **2**(2–3), p. 363–388 (2001)
20. Cornwell, J., et al.: Affordance Theory for Improving the Rapid Generation, Composability, and Reusability of Synthetic Agents and Objects. 12th Conference on Behavior Representation in Modeling and Simulation (2003)
21. Ali, W., Moulin, B.: 2D-3D Multi-Agent Geo-Simulation with Knowledge-Based Agents of Customers' Shopping Behavior in a Shopping Mall. In: Cohn, A.G., Mark, D.M. (eds.) Spatial Information Theory, p. 445–458. Springer, Berlin (2005)
22. Frank, A.: Spatial Communication with Maps: Defining the Correctness of Maps Using a Multi-Agent Simulation. In: Freksa, C., et al. (eds.) Spatial Cognition II – Integrating Abstract Theories, Empirical Studies, Formal Methods, and Practical Applications, p. 80–99. Springer, Berlin, Heidelberg, Germany (2000)
23. Hochmair, H., Lüttich, K.: An Analysis of the Navigation Metaphor – and Why it Works for the World Wide Web. Spat. Cogn. Comp. **6**(2), 235–278 (2006)
24. Lovas, G.: Modeling and Simulation of Pedestrian Traffic Flow. Trans. Res. B. **28**(6), 429–443 (1994)
25. Pelechano, N., Badler, N.: Modeling Crowd and Trained Leader Behavior during Building Evacuation. IEEE Computer Graphics and Applications **26**(6), 80–86 (2006)

Wayfinding Behavior

► Wayfinding: Affordances and Agent Simulation

Wayfinding, Landmarks

STEPHEN C. HIRTLE
School of Information Sciences, University of Pittsburgh, Pittsburgh, PA, USA

Synonyms

Anchor points; Cognitive mapping; Navigation aid; Organizing concept; Trichotomous theory of landmarks; Survey knowledge; Typology of landmarks

Definition

A landmark is a prominent element or location in an environment that serves to define the location of other elements or locations. A landmark can serve as an organizing feature or as a navigational aid and can be characterized by its visual, structural and semantic features [1]. Landmarks, or anchor points, are used in two fundamentally different ways [2]. First, a landmark can be used as an organizing concept for an environment. In this way, a landmark can be used in part to define a neighborhood. Second, landmarks can serve as navigational aids by providing choice points along a route of travel, confirmation of the correct route of travel, verification of arrival at the destination, or evidence of the present orientation in the case of distal landmarks.

Landmarks are set apart from non-landmarks by a number of defining characteristics that include visual characteristics, such as singularity and prominence of the landmark in contrast to surrounding locations [3], semantic characteristics, such as content, meaning, use or cultural significance, and structural characteristics, such as accessibility and the centrality in transportation network. Landmarks may be local landmarks that you navigate to and from or distant landmarks that you use as a general referent to orient directional movement [2,3]. Each of these characteristics can be quantified for automated inclusion into wayfinding systems.

Historical Background

In a pioneering work, Kevin Lynch [3] argued that the cognitive map of a city environment was defined by five types of elements: paths, edges, districts, nodes and landmarks. Landmarks according to Lynch ranged from local cues, such as signs, to distant cues, such as mountains, as long as they were visually distinctive.

Siegel and White [4] presented the most influential theory relating landmarks to wayfinding in a trichotomous theory of landmarks, route and configurational knowledge. According to Siegel and White, individuals learning a new space will first acquire knowledge of specific landmarks, followed by knowledge of the routes between landmarks. With enough experience with a space, one finally acquires configurational or survey knowledge of the environment, which allows for an understanding the relative orientation of landmarks within a 2-dimensional space.

There have been numerous refinements and challenges over the years to both Lynch's typology and Siegel and White's original formulation [5]. There is some evidence that landmark and route knowledge are acquired in parallel rather than sequentially, while others argue that the acquisition of survey knowledge requires more than just experience [1]. There is also evidence that route and survey knowledge may be mediated by different neural structures [6].

Scientific Fundamentals

There are several attempts in the literature to quantify what objects should be used as landmarks to assist with the automatic generation of landmarks for navigational systems. The exact choice of landmarks will depend in part on the mode of transportation with most of the current research on landmarks for either vehicle drivers or pedestrians [7,8]. The extraction of landmarks from the environment also depend on the image schemata that is used by the navigator with indoor navigation more often dependent on signage and the flow of other pedestrians.

In large survey of respondents in Columbus, Ohio, it was observed that half the locations listed as best known and most familiar were common with other respondents, while the other half reflected individual activity patterns. In a describing pedestrian routes in Venice, Italy, individuals used landmarks that were both on the route, such as streets, bridges, and plazas, near the route, such as distinctive buildings or prominent signs. Children have been known to use transient items as landmarks (e. g., *turn left at the house with the large red ball in the front yard*) and travelers often need to spend some effort to extract distinctive characteristics in a foreign environment.

The typology of landmarks has included landmark identity, location, and dominance (with regard to the surrounds), while landmark use has included landmarks as choice points, origin or destination, orientation along a route, regional differentiating feature, home base for path integration vectors, and priming features influencing expectations [2].

Key Applications

Wayfinding Systems

Automated wayfinding systems are the most common application for the generation and display of landmarks. While human generated directions are often based on landmarks (e. g., *turn right just past the movie theatre*), most automated systems do not use local landmarks other than for the final destination. Landmarks are particularly difficult to determine given the large number of possible landmarks. However, several recent studies have developed algorithms for measuring "landmarkedness" of a location [9]. In this case, the dimensions of a landmark were taken from earlier work [1], which classified landmarks according to their semantic, structural, and visual attributes. Their analysis of characteristics, such as size, color, building shape, cultural importance and signage, showed that each of these components were important in the selection of buildings to be used as landmarks in the instructions for a given path. Caduff and Timpf use a diagrammatic approach that selects spatial cues based on orientation and distance to the landmark along with the salience of the potential landmark objects. Elias used data mining to determine the relative uniqueness of visual characteristics among the pool of potential landmarks using unsupervised learning methods [10].

Urban Design

To the extent that landmarks are an important determiner of the navigability of a city [3], then using the landmarks to improve the design of urban environments is a natural extension of the research on landmarks [11]. Homogenous, non-differentiated spaces are going to be particularly difficult to navigate. Furthermore, the uniqueness of landmarks is culturally dependent. Redundant artifacts that define a the nature of particular city can be useful landmarks (a metro stop in Paris or the corner dairy shop in New Zealand), as can the artifacts that unique, in terms of geometry, symbolism, or even legend (such as *Balcone di Guilietta* in Verona, Italy).

Virtual Environments

A number of researchers have shown that the inclusion and use of landmarks in virtual environments can improve navigation [12,13]. Related work in robotic navigation is often dependent on landmarks, as well.

Electronic Navigation

The dissertation research by Sorrows found that landmarks on the Web have a variety of visual, semantic and structural characteristics, and are used as key elements in navigation, which are recalled easily [14]. This research involved evaluating web page elements such as the number of links to the page, the amount of text and graphics on the page, the content of the text, and the number of link choices on the page. The research showed that pages that were easily recalled in a series of path navigation questions had higher value on the landmark quality measure than the general set of pages from the web site.

Future Directions

The explosion of interest for developing in-car navigation systems with multi-media displays provides a fertile ground for future work on landmarks. Most likely the usability of such systems will be greatly improved by providing schematic information with selective landmarks. In addition, systems will improve by matching the information in the database used for routing with the signs and landmarks that are found in the environment. Such systems should be tied to an audio output so that visual attention is not taken away from the primary task of monitoring traffic conditions on the road.

Other important developments will most likely occur in developing appropriate wayfinding theories for intermodal transportation, such as moving from bicycle to bus to subway and navigation through three-dimensional environments, such as subway stations. There is also active research programs in exploring spatial cognition and the use of landmarks in special populations, such the elderly or visually impaired travelers [15].

Finally, an increase in the use of intelligent sensors, such as RIFD and GPS, will likely have dramatic effects on wayfinding technologies.

Cross References

▶ Privacy Preservation of GPS Traces

Recommended Reading

1. Sorrows, M.E., Hirtle, S.C.: The Nature of Landmarks for Real and Electronic Spaces. In: Freksa, C., Mark, D. (eds.) Spatial Information Theory, pp. 37–50. Springer, Berlin (1999)
2. Golledge, R.G.: Human wayfinding and cognitive maps. In: Golledge, R.G. (ed.) Wayfinding behavior: Cognitive mapping and other spatial processes, pp. 5–45. Johns Hopkins Press, Baltimore (1999)
3. Lynch, K.: The image of the city. MIT Press, Cambridge, MA (1960)
4. Siegel, A.W., White, S.H., The development of spatial representations of large-scale environments. In: Reese, H.W. (ed.) Advances in Child Development and Behavior, pp. 9–55. Academic Press, New York (1975)
5. Allen, G.L.: Spatial abilities, cognitive maps, and wayfinding: Bases for individual differences in spatial cognition and behavior. In: Golledge R.G. (ed.) Wayfinding behavior: Cognitive mapping and other spatial processes, pp. 46–80. The Johns Hopkins University Press, Baltimore (1999)
6. O'Keefe, J., Nadel, L.: The Hippocampus as a Cognitive Map. Clarendon Press, Oxford (1978)
7. Ishikawa, T., Montello, D.R.: Spatial knowledge acquisition from direct experience in the environment: individual differences in the development of metric knowledge and the integration of separately learned places. Cognit. Psychol. **52**, 93–129 (2006)
8. Michon, P.E., Denis, M.: When and Why Are Visual Landmarks Used in Giving Directions? In: Montello, D.R. (ed.) Conference on Spatial Information Theory: Foundations of Geographic Information Science, pp. 292–305. Springer-Verlag, Berlin (2001)
9. Nothegger, C., Winter, S., Raubal, M.: Selection of salient features for route directions. Spatial Cogni. Comput. **4**, 113–136 (2004)
10. Elias, B.: Extracting Landmarks with Data Mining Methods. In: Kuhn, W., Worboys, M., Timpf, S. (eds.) Spatial Information Theory: Foundations of Geographic Information Science, pp. 375–389. Springer-Verlag, Berlin (2003)
11. Portugali, J.: Self-Organization and the City. Springer, Berlin (2000)
12. Darken, R.P., Peterson, B.: Spatial Orientation, Wayfinding, and Representation. In: Stanney, K.M. (ed.) Handbook of Virtual Environments: Design, Implementation, and Applications, pp. 493–518. Lawrence Erlbaum Associates, Mahwah NJ (2002)
13. Abowd, G.D., Mynatt, E.D., Rodden, T.: The human experience. IEEE Pervasive Comput. **1**, 48–57 (2003)
14. Hirtle, S.C., Sorrows, M.E.: Navigation in electronic environments. In: Allen, G. (ed.) Applied Spatial Cognition, pp. 103–126. Lawrence Erlbaum Associates, Mahwah NJ (2007)
15. Allen, G.: Applied Spatial Cognition: From Research to Cognitive Technology. Lawrence Erlbaum Associates, Mahwah NJ (2007)

WCS

▶ deegree Free Software
▶ OGC's Open Standards for Geospatial Interoperability

Weather Analysis

▶ Weather Information, Surface Transportation

Weather Forecasting

► Weather Information, Surface Transportation

Weather Information, Surface Transportation

JOHN J. MEWES[1], STEPHEN G. GADDY[1], ROBERT D. HART[1], LEON F. OSBORNE[1,2]
[1] Meridian Environmental Technology, Inc., Grand Forks, ND, USA
[2] Regional Weather Information Center, University of North Dakota, Grand Forks, ND, USA

Synonyms

Weather analysis; Weather forecasting

Definition

Surface Transportation Weather is a general term referring to weather analysis and forecasting services customized toward surface transportation industries. Surface transportation itself is a generalized term usually referring collectively to transportation by road, rail, water or pipeline, as well as the ground operations of the aviation industry. The operation of these industries can be severely impacted by adverse weather conditions. As such, successful management of these industries often requires highly specialized, location- and time-specific weather consulting services in order to improve transportation safety, efficiency and reliability by planning for these adverse weather conditions.

Surface Transportation Weather is by definition a cross-disciplinary industry. Persons providing services in this industry generally have some level of expertise in both weather and one or more surface transportation industries. The recipients of these services range from surface transportation industry management to the traveling public.

Historical Background

The need for Surface Transportation Weather evolved partly due to the maturation of the surface transportation infrastructure and partly due to technological advances in meteorology that occurred during the 20th century. Aviation had a significant impact on the development of meteorology during this period, forcing the weather community to rapidly enhance data communications, analyses of surface and upper air conditions, and techniques to monitor and predict conditions both on the earth's surface and within the atmosphere. The expansion of aviation infrastructure that drove the meteorological requirement was paralleled by a rapid growth of the highway system. Although the rail and marine transportation networks were well established during this period, it was arguably the convergence of the tremendous technological strides in meteorology and the growing demands associated with highway travel that led to the initiation of the Surface Transportation Weather industry.

Paved roads facilitated the expansion of highway travel. It offered greater mobility than dirt and gravel surfaces; however, bare pavement became the expectation of those who used these paved roads. Government developed maintenance divisions to manage the highway infrastructure and assure that adverse conditions did not reduce the road's safety or impact the user's mobility. A critical responsibility was maintenance during winter conditions, leading to a new industrial sector focusing upon the removal from and prevention of snow and ice on roadways.

A key catalyst in the development of weather support services for highway and runway operations was the introduction of a pavement sensor that measured pavement temperature, surface condition, and the chemical concentration on the top surface of the pavement. For winter maintenance these elements were critical and the inclusion of this data made existing weather information more complete and useful for the maintenance provider. As a result, composite weather and pavement condition measurement systems proliferated and became known as Road/Runway Weather Information Systems (RWIS). The RWIS concept was introduced around 1970 in Europe and North America as independent monitoring systems. The number of individual systems has grown rapidly and the stand-alone systems have now been integrated into extensive networks throughout areas impacted by snow and ice. There are nearly 1500 RWIS sites in North America and nearly double that worldwide.

The expansion of RWIS networks fostered a new approach to winter maintenance by permitting maintenance personnel to know the observed pavement temperature and pavement conditions throughout their area of responsibility. In the mid-1980s this observational network was augmented with the introduction of pavement specific forecasts that used energy and mass balance models to create projections of the pavement conditions 24 to 48 hours into the future. These forecasts were initially provided primarily by manufacturers of RWIS networks as a value-added service. Over time, research in the area of winter maintenance came to show a great deal of value in proactive (as opposed to reactive) snow and ice control, referred to as 'anti-icing'. Implementation of anti-icing operations requires very precise forecasts of precipitation timing, amounts, types and pavement temperatures. As a result of this anti-icing movement and the increasing expectations of the traveling public, the demand for weather and pavement forecasting and analysis services grew. Over time the other sectors of today's Surface Transportation Weather industry (including the traveling public) also came to see the weather as something that could be planned for and managed around. The growth in the need for these weather support services has allowed Surface Transportation Weather to become a distinct and thriving industry.

Scientific Fundamentals

The Surface Transportation Weather industry generally fills two key requirements that set it apart as its own industry. The first requirement is the need for highly time- and location-specific weather information. This information is typically generated using a combination of computer-based weather analysis and/or prediction models and expertise from meteorologists trained in analyzing and forecasting weather as it pertains to a specific surface transportation industry. The second requirement is to apply this weather information in the framework of the affected surface transportation industry. An example of this would be the application of a pavement model to analyze and/or predict the response of a roadway to the combination of weather conditions, traffic, and actions taken by a highway maintenance agency.

Weather Prediction

As mentioned, the first requirement of highly time- and location-specific weather information is often at least partially fulfilled through the use of computer models designed to predict the expected evolution of the weather over time and space given observations of (recent) past and present weather conditions. These models are commonly referred to as numerical weather prediction (NWP) models. Weather prediction is an initial- and boundary-value problem, meaning that its solution is dependent both upon the initial state of the atmosphere in the model as well as the boundary conditions that are applied. The initial state of the atmosphere is typically diagnosed within the model by gathering and integrating weather observations from numerous sources. These sources include government weather centers, airport surface and upper-air observations, weather radars and satellites, and a host of other observing platforms available globally. It is not uncommon for affected industries to deploy their own localized observing networks ('mesonets') to support weather services provided on their behalf. A good example of this are the Environmental Sensor Stations (ESS) frequently deployed to collect real-time observations of weather conditions in the RWIS networks discussed above.

Numerical weather prediction models are normally equipped with packages for assimilating these broad-ranging observations to arrive at a snapshot of the three dimensional state of the atmosphere at a given time. This snapshot of the atmosphere is then applied as the 'initial value' in the atmospheric prediction problem. Assimilation is a word commonly used to refer to the process of re-mapping and integrating observations (typically multi-source and geographically diverse) together into the regularized structure of the model in a manner consistent with principles of geographical information systems (GIS) and the laws of physics that govern the atmosphere. The assimilation process often relies partially upon previously generated forecasts in recognition of the time-weighted value of observations taken at earlier times in different locations. The use of this 'first guess' field can help to compensate for holes in the global observing network and improves forecast consistency from one model run to the next.

The other half of the weather prediction problem pertains to what are called 'boundary conditions'. Unless the weather prediction model is configured to run globally, a prospect that generally requires immense computational resources, there are always horizontal boundaries to the simulation at which assumptions must be made. The most common technique is to apply simulated conditions from a larger scale model (such as those run at some of the larger governmental modeling centers of the world) at those boundaries. Because of this requirement, it is desirable that the domain that is configured for a given numerical weather prediction problem be large enough that the area of primary interest is well removed from the horizontal boundaries of the domain. Since weather systems are transient in the atmosphere, this generally requires that the size of the simulation domain increase as the length of the desired simulation increases.

In addition to these horizontal boundary conditions, there are also vertical boundary conditions that must be applied. In simple terms, the boundary condition that is usually applied at the top of the simulation domain is that there be no motion through this boundary. The lower boundary, which is the Earth's surface, requires much more sophisticated handling within the model. This usually involves the application of a land surface model (LSM) which is coupled to the atmospheric model that simulates the exchanges of heat, moisture, and momentum between the Earth's surface and the atmosphere. This process normally requires detailed information on vegetation and terrain characteristics and how they vary over the model's spatial domain.

Given the initial and boundary values, the atmospheric prediction problem becomes one of mathematical integration of a system of non-linear differential equations that govern the evolution of the atmosphere. These equations are commonly referred to as the Navier-Stokes equations. The solution of these equations at a given time provides the expected state of the atmosphere at a short time thereafter. The continued solution of these equations at successive times leads to a prediction of the evolution of the entire atmosphere over time.

Computer models typically store snapshots of the state of the atmosphere at regular intervals throughout a simulation. These snapshots generally serve as the starting point for the creation of most weather forecasts in the world today. However, how the snapshots from the computer model get transformed into a forecast for a specific industry may vary substantially. In some cases this model output is provided directly to the affected industry, or at least to an application which applies the weather forecast directly to an industry problem. In other cases the output of one or more computer models will be analyzed and adjusted by a trained meteorologist, then converted into either written or spoken dialogue to end-users within the affected industry. A middle-ground exists between these two extremes wherein trained meteorologists use digital forecast editors to modify and/or blend forecast information from multiple sources. This process allows the highly detailed model output from the computer simulation to pass through to the end-user or other applications, but also has the benefit of a meteorologist's adjustments to the forecast (which are often required in order to ensure a consistently reliable service).

Weather Analysis

Not all weather services provided to surface transportation industries involve weather prediction. In many cases the service is focused upon the provision of real-time synopses and warnings of present weather conditions. For example, travelers may desire knowledge of severe weather conditions in their immediate vicinity so they can alter their travel plans. Train derailments may result from severe weather conditions that are often identifiable in real-time and predictable shortly in advance of occurrence. Departments of Transportation, and other road maintainers, may wish to receive alerts as certain events occur (such as pavement temperatures dropping to freezing on non-dry surfaces) such that they can react quickly to unforeseen events during typical off-hours without the need to continually monitor the situation at all times.

Although useful in some cases, the confined nature of many severe weather events makes traditional in-situ observing networks of limited value in this process. The high cost of deploying these networks typically prevents siting densities from being adequate to sample all severe weather events. A common exception to this rule are the specialized mesonets often deployed by specific industries to better sample weather conditions of special interest to each industry. For example, a highway maintenance agency may place an ESS at a common trouble spot (such as a mountain pass) as part of their RWIS network.

For industries with a large geographical coverage area and unpredictable exposures to severe weather, techniques for remotely sensing weather conditions are often of most value. In particular, Doppler and/or Polarmetric weather radars can provide high-resolution observations of wind and/or precipitation occurrences that require special caution on the part of an affected industry. For example, the provision of weather services to railroads often involves real-time monitoring of Doppler weather radar data for wind speeds that could threaten a train derailment. If detected, it is not uncommon for trains to be halted out of harm's way until the threat has passed. Another very serious concern of all land-based surface transportation industries is flooding. Prediction of flooding at a specific location well in advance of its actual occurrence is very difficult due the inherent unpredictability of the weather conditions that lead to it. However, since most occurrences of flooding are preceded by heavy precipitation events that can be observed in great detail by weather radars, it is possible to predict (with a short lead time) which locations may suffer severe flooding conditions by applying detailed estimates of precipitation from weather radars in combination with hydrologic models capable of predicting the flow of this water after it reaches the Earth's surface. Armed with this knowledge transportation management officials can close roads or keep trains away from sections of railroad where track integrity may be compromised.

Industry Application

Historically, much of the 'industry application' side of the Surface Transportation Weather industry consisted primarily of a person trained in both weather and industry sciences interpreting weather information in the context of the industry operations. However, as the amount of reliable detail in weather information has increased over time, so have the number of applications that build upon this detail to make its application more of a science. Today Surface Transportation Weather involves the combined application of industry-specific GIS databases and sophisticated models to transform time- and location-specific weather information into industry specific decision support tools.

In the highway winter maintenance industry this might involve the application of a pavement model to better understand the implications of expected weather conditions for the safety and mobility of roads within an agency's highway system. Pavement modeling typically involves modeling heat and moisture balances at the surface of the roadway. This includes modeling of radiative, sensible, and latent heat fluxes, including those associated with falling precipitation, runoff, condensation/deposition, evaporation/sublimation, and phase changes of moisture residing atop the roadway. It also includes modeling of the flows and storage of energy into and out of the pavement substrate, a process which can be done more accurately if detailed construction profiles for a particular stretch of road are available in GIS databases.

In recent years the integration of agency GIS databases into Surface Transportation Weather services has been furthered by the development of Maintenance Decision Support Systems (MDSS). MDSS represent the next step in the evolution of Surface Transportation Weather services for the highway maintenance industry by using decision logic software to convert expected weather and pavement conditions into a suggested plan of action for roadway maintenance. In some cases this involves modeling the impact these actions will have upon the layer of liquid water, snow, ice, and deicing chemicals (the 'dynamic layer') that the agency desires to stabilize or remove from the roadway. In other cases the course of action is drawn from a library of practices other agencies have used successfully in similar situations in the past. In general, the provision of MDSS services has necessitated a tighter integration of Surface Transportation Weather services with maintenance agency GIS databases. For example, the removal of slush and chemicals from a roadway is heavily dependent upon traffic considerations on a road segment. As such, automobile and truck traffic volumes as a function of time of day for a particular road segment might be extracted from agency databases in order to support modeling of this process. Other GIS attributes such as the characteristics of the roadway environment, the extents of maintenance routes, the level of service, and the chemical mixtures available for fighting snow and ice are also regularly tapped during the provision of MDSS services.

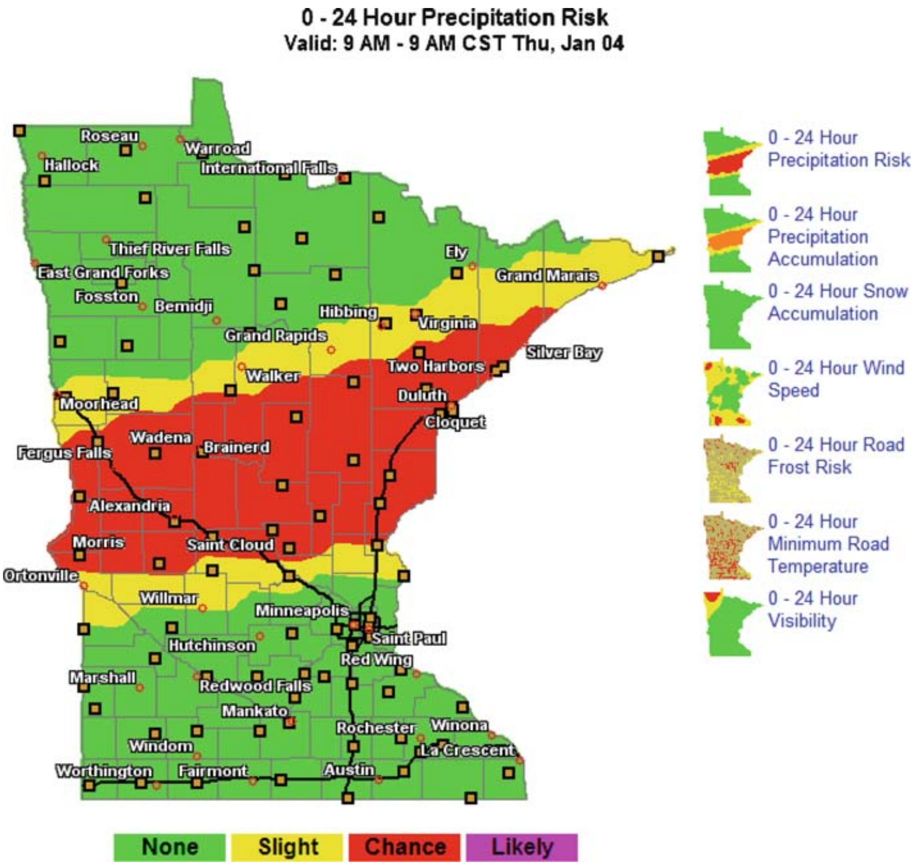
In addition, snowplows are now being routinely equipped with GPS-based vehicle tracking and reporting devices which can feed MDSS and traveler information applications with critical real-time GIS information on what maintenance has been performed on a given stretch of road (and when) as well as road and weather conditions encountered at that time. While developed for one-way transfer from snowplows to agency databases, these systems are now advancing into two-way systems in which the applications receiving data from the trucks not only feed information to those at fixed computer workstations, but also back into the snowplows, resulting in a more complete mobile operations center. This information-on-the-go is critical in many states where manpower and funding limits the ability to keep decision-makers in the office, requiring critical decisions to be made by those out clearing the roads.

In rail, one industry application involves the cross referencing of GIS databases describing railroad locations and exposures with analyses of the short-term potential for excessive wind speeds created through analysis of Doppler weather radar data and other supporting information. The goal of such an application is to reduce the risk of train derailment by strong crosswinds, an extremely costly and not uncommon occurrence. Another application involves the provision of guidance as to where and when a railbed might have been compromised or destroyed by flood waters. This can be accomplished by application of high-resolution precipitation estimates (usually from weather radars) in hydrologic models that are capable of simulating the resulting runoff patterns, and cross referencing these patterns with the locations and drainage characteristics of existing railroads.

A common industry application in the marine industry is the operation of a wind wave model to predict problematic or dangerous wave heights. Such models work similarly to atmospheric prediction models, but instead predict the response of oceans and waterways to forcing from the wind given bathymetrical (sea floor topography) characteristics. Different considerations are commonly made in these models depending upon whether shallow or deep waters are of interest. When cross referenced with GIS data on shipping routes, these models can alert marine vessels of conditions that may warrant pause or alteration of course.

Examples of Geospatial Relationships and Presentations

Surface Transportation Weather and the foundation framework of weather analysis and weather prediction are inherently geospatial in nature. While only recently embracing many of the concepts of traditional GIS in its applications, weather analysis and prediction has always required application of geo-referencing concepts. The source of input data are expressed in geographical locations during the process of assimilation with the results typically expressed in a conformal mapping system for input into weather prediction models. Output from these models is mapped back to geographical locations to provide a place-based forecast of future weather conditions. The evolution of surface transportation in recent years has incorporated broader GIS concepts to integrate across weather and non-weather thematic relationships. This is typically utilized the most during final application product generation when the fusion of themes provides the information content specific to the end-user's needs. Most often this fusion of themes results in a static graphic that is available for the end-user to incorporate in their decision making process. For example, the



Weather Information, Surface Transportation, Figure 1 Thematic views of various forecast parameters selectable within a web-based context. Weather forecast themes are fused with geopolitical, transportation network, and city themes to produce an end-user product supporting Surface Transportation Weather applications

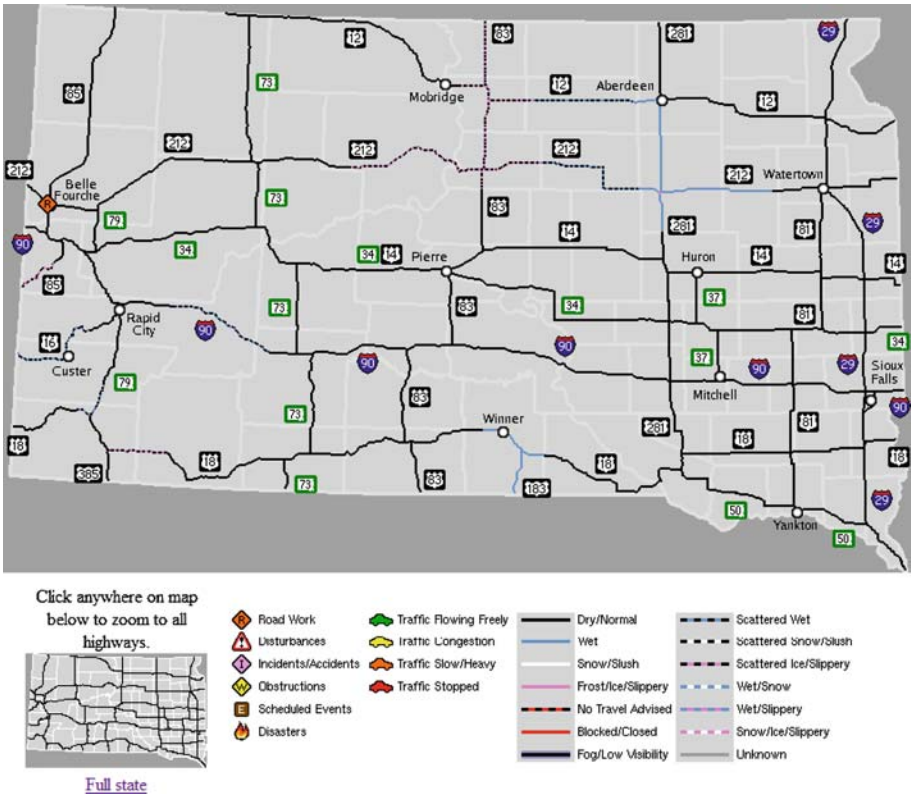
generation of web-based Surface Transportation Weather products provides the end-user the capability of monitoring the threat of frost on roads across a statewide road network (Fig. 1). By selecting different themes the end-user can quickly view the results of the Surface Transportation Weather forecasts depicted in the end-user's desired geospatial context.

In some Surface Transportation Weather applications, the results of the weather become the driving force for the end-user's use of information. Figure 2 is an example of road conditions resulting from weather events. This application fuses the geopolitical boundary and the agency road network with the current road conditions and the site-specific impacts these road conditions are having on the traffic flow, including the occurrence of traffic incidents. The construction of this application utilizes agency road network, incident, and road condition GIS databases.

The growth in information technology capabilities within transportation agencies has also permitted the utilization of interactive desktop decision support tools that more fully incorporate GIS concepts. The most sophisticated of these at present are the Maintenance Decision Support Systems described above. These are generally client-side applications that provide interactive selection of viewable themes (Fig. 3). All displayed data are geo-coded and displayed thematically. Selection of themes include point-based data (real-time maintenance vehicle locations, weather and road condition observations, locations of camera images, and landmarks), raster images (weather radar, weather satellite, analyses and predictions for various weather variables), and lines/polygons (road maintenance routes, road networks, political boundaries, and land-surface features). Navigation (pan and zoom) features may be provided to permit the user to select focused areas, while the themes that are viewed may be chosen through a combination of icons and/or pull-down menus.

Key Applications

The science and GIS principles behind the key applications of Surface Transportation Weather services have been discussed above. These applications generally offer the affected industries the prospects of improved operating efficiency, safety and/or mobility.



Weather Information, Surface Transportation, Figure 2 The depiction of weather impacts on a highway system is generated through the joining of road condition, traffic, and incident databases with the agency's road network

Road and Airport Maintenance and Operations

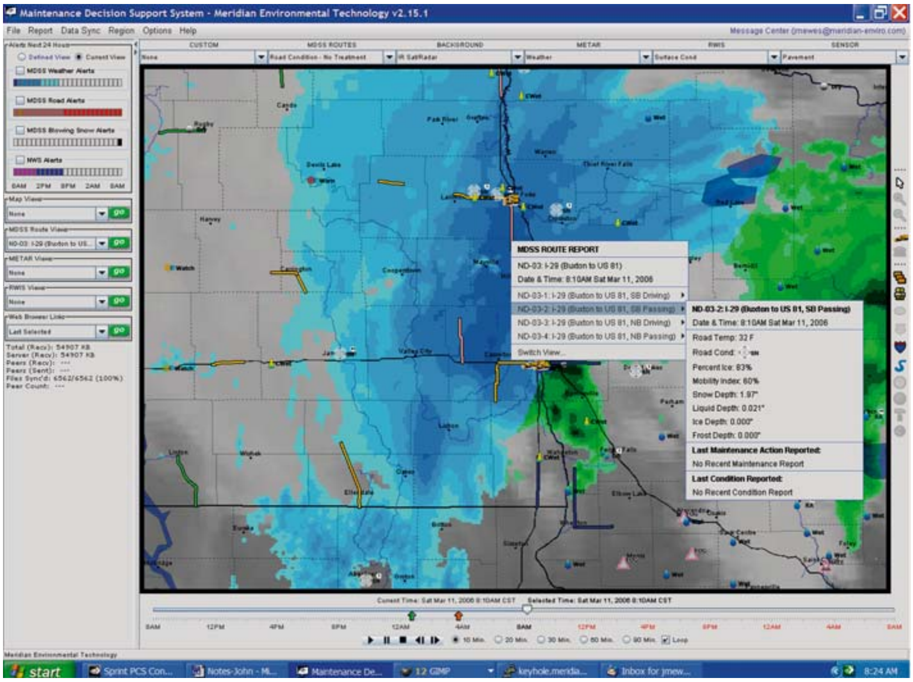
In highway winter maintenance and airport ground operations, Surface Transportation Weather services help to optimize crew scheduling by better anticipating where and when maintenance personnel and resources will be required. Applications such as MDSS also offer the prospect of optimized chemical usage, helping to ensure that the amount of chemical dispersed is adequate from a safety and mobility viewpoint but not overdone to a point that it would adversely impact both agency budgets and the environment. Knowledge of expected weather conditions can also help highway agencies avoid costly weather-related damages to road construction projects. Airport ground operations can likewise benefit from this knowledge by planning for difficulties normally associated with impending weather conditions, or by keeping ground personnel indoors during lightning.

Rail Industry

In the rail industry, Surface Transportation Weather services help to reduce the risk of derailment by keeping trains out of potentially dangerous crosswinds due to severe thunderstorms, funneling through mountain passes, major winter storms, and hurricanes. They may also help to reduce the risk of derailment caused by railroad buckling due to expansion/contraction under conditions of excessive heat/cold. Weakening or complete erosion of the railbed by floodwaters is also a major concern of the railroad industry. Joint application of high-resolution radar estimates of rainfall and a hydrologic model for predicting runoff behaviors, coupled with industry databases on railroad locations and drainage characteristics, can help to identify locations of potential washouts in the rail system.

Marine Transportation

In the marine industry, Surface Transportation Weather services help to predict winter storms and hurricanes which can create hazardous situations for both cargo and industry personnel (including those in ports). The prediction of wind wave heights commonly provided in these services may reduce the risk of capsize in small watercraft, while helping larger vessels plan alternate routes and/or secure cargo. The prediction of conditions of low visibility can also help to alter port and waterway operations to diminish the threat of collision. Surface Transportation Weather services can also provide guidance as to where dangerous ice may be encountered, whether from a waterway freezing or icebergs moving through ocean routes.



Weather Information, Surface Transportation, Figure 3 An example of a Maintenance Decision Support System client-side graphical user interface that incorporates selectable themes to display various data types. Drop-down lists on top provide various weather-centric data including point and raster themes. Icons to the right of the map provide themes corresponding with transportation, hydrography, and city themes

Traveler Information

A burgeoning Surface Transportation Weather service geared toward the traveling public is the provision of detailed route- and time-specific weather information through traveler information services, such as the 511 systems of the United States. The goal of such services are to provide travelers highly specific road condition, construction, traffic, and weather information in order to help them better plan for travel conditions or possibly alter those travel plans. The provision of such services generally involves the cross referencing of GIS databases describing the location of distinct road segments and other landmarks with high-resolution, short-term digital weather forecasts. The goal is to provide travelers with the level of detail required to make prudent decisions about how, where and when they travel.

Future Directions

Cross-disciplinary environmental sciences such as Surface Transportation Weather will likely see substantial changes in the near future as the industry's ability to accurately diagnose and predict weather conditions on very fine scales continues to improve. Weather is increasingly viewed as something that can be planned for and carefully managed around rather than as an inevitable drag on affected industries. As these cross-disciplinary applications grow in complexity the role of GIS in predicting and managing weather's impacts will also grow.

Recommended Reading

1. Daley, R.: Atmospheric Data Analysis. Cambridge University Press (1996)
2. Kalnay, E.: Atmospheric Modeling, Data Assimilation and Predictability. Cambridge University Press (2004)
3. Vieux, B.E.: Distributed Hydrologic Modeling Using GIS. Kluwer Academic Publishers (2004)
4. Dikau R., Saurer, H.: GIS for Earth Surface Systems. Gebruder Borntraeger (1999)
5. Perry, A.H., Symons, L.J. (ed.): Highway Meteorology. E & FN Spon (1991)
6. Minsk, L.D.: Snow and Ice Control Manual for Transportation Facilities. (1998)
7. Thiebaux, H.J., Pedder, M.A.: Spatial Objective Analysis with Applications in Atmospheric Science. Academic Press (1987)

Web Application Hybrids

► GIS Mashups

Web Application Server

► Web Mapping and Web Cartography

Web Coverage Service

► deegree Free Software
► OGC's Open Standards for Geospatial Interoperability

Web Feature Service (WFS)

ABHINAYA SINHA
Department of Computer Science and Engineering,
University of Minnesota, Minneapolis, MN, USA

Synonyms

Geographic markup language; GML; WFS; Web map service; WMS

Definition

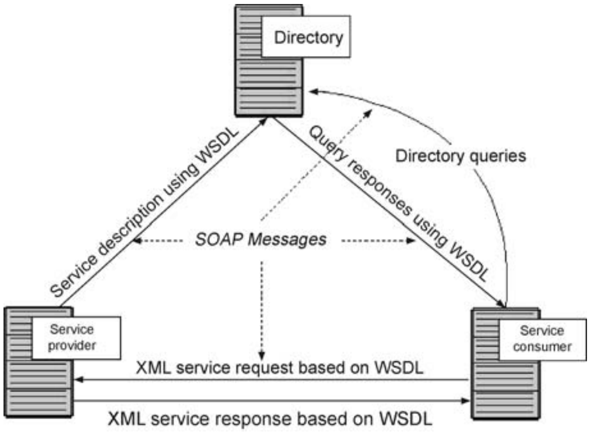
The Web Feature Service (WFS) is an interface specified by the Open GIS Consortium (OGC) that allows for the exchange of geographic data across the Web. It defines the rules for requesting and retrieving geographic information using the Hyper Text Transmission Protocol (HTTP). The interface describes the data manipulation operations on geographic features. Extensible Markup Language (XML) -based Geographic Markup Language (GML) is used for exchange of information. It should be noted that WFS supports the vector data model.

Historical Background

The Open GeoSpatial Consortium, Inc (OGC) has been instrumental in the formulation of the Web Feature Service. OGC is an international industry consortium of companies, government agencies and universities participating in a consensus process to develop publicly available interface specifications. Regarding Web Services, OGC has specified web interfaces based on a model supporting request and response rules using Hyper Text Transmission Protocol (HTTP) and XML. The web feature service specification was first introduced in May 2002 when OGC released WFS 1.0. The latest version, WFS 1.1., was released in May 2005. Updates could be frequent as and when changes are made.

Scientific Fundamentals

Web Services is any software that makes itself available over the Internet and allows standard XML messaging protocol such as SOAP (Simple Object Access Protocol) as a messaging system. Since web services are based on simple and non-proprietary standards, they promote interoperability as the transmitted information can be understood by machines independent of their underlying operating system, programming languages, development environments or their location The main idea of a web service is that it is possible for machines to publish information about the kind of services they offer and the parameters that need to be passed to them so that they can provide the promised services. A web service is based on a service-oriented architecture, as shown in Fig. 1. There are three fundamental operations: publish, find and bind. Service providers publish services to a service broker. Service requesters find the service and bind to them. This concept can be applied to geographic data. Hence, WFS is a specification whereby one machine (say, machine A) can request some other machine (say, machine B) for geographic information. The reason the machine requests information is that it does not have that information itself and hence must resort to some other machine to provide that information. Once A has the information it needs, it is now possible for A to use that information according to its requirement. This is possible because the supplied information is encoded in XML-based GML and hence, is machine independent. Since geographic data is being referred to in this case, A would normally use this information to construct a map or add additional layers to a map that it already was capable of rendering. This is the most common way in which the services provided by a web feature server may be used. However, there are other ways of utilizing the web feature service.

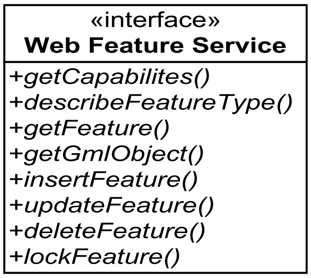


Web Feature Service (WFS), Figure 1 Web Service and its Service-Oriented Architecture (courtesy: Mapping Hacks by Schuyler Erle, Rich Gibson, and Jo Walshs)

Features of WFS

WFS provides interfaces for describing data manipulation operations on geographic features. Data manipulation includes:

1. Creating a new feature instance
2. Deleting a feature instance
3. Updating a feature instance
4. Locking a feature instance
5. Getting or querying features based on spatial and non-spatial constraints.



Web Feature Service (WFS), Figure 2 Operations supported by a WFS

The first four interfaces have to do with supporting transactions while the last one is concerned with retrieving geospatial data.

Figure 2 shows the operations defined to support transaction and query processing, in the form of a UML interface diagram.

These operations can be described as follows:

1. **GetCapabilities**
   The web feature service should describe its 'capabilities', that is, the feature types that the WFS can service as well as the operations defined on each of those feature types. When this request is made, the web feature server must return an XML document describing the feature types that it can service and the operations defined on each on them. An example of a GetCapabilities request is given next:
   http://www2.dmsolutions.ca/cgi-bin/mswfs_gmap?SERVICE=WFSVERSION=1.0.0REQUEST=getcapabilities
2. **Describe Feature Type**
   A Web Feature Service must be able to describe the structure of any feature type that it can service. The following example requests the schema description of the feature type TreesA_1M. http://www.someserver.com/wfs.cgi?SERVICE=WFSVERSION=1.1.0REQUEST=DescribeFeatureTypeTYPENAME=TreesA_1M
3. **GetFeature**
   A WFS returns the GML data associated with the features specified in this request. This is the request that is used by a client to get the geodata associated with features supported by the web feature server. The example below gets geographic data associated with parks:
   http://www2.dmsolutions.ca/cgi-bin/mswfs_gmap?SERVICE=WFSVERSION=1.0.0REQUEST=getfeatureTYPENAME=park
4. **Transaction**
   A web feature service may be able to service transaction requests. Note that this feature is optional. Even if the service does not provide this capability, it still is a web feature service. Transaction requests are concerned with operations that modify features on the server such as create, update and delete operations on geographic features. The following example deletes all feature instances identified by "RoadL_1M.1013"
   http://www.someserver.com/wfs.cgi?SERVICE=WFSVERSION=1.1.0REQUEST=Transaction
5. **GetGMLFeature**
   A web feature service may, optionally, be able to service a request to retrieve element instances by traversing XLinks that refer to their XML IDs. The requirement of the client is that it must specify whether nested XLinks embedded in returned element data should also be retrieved.

Types of Web Feature Services: It is not necessary that a Web Feature Service provide support for all the above operations. Accordingly, Web Feature Services can come in three flavors: Basic WFS, XLink WFS and Transactional WFS.

These three types of WFS are explained as follows:

Basic WFS: A basic WFS only allows querying and retrieval of features. That is, it supports the GetCapabilities, GetFeatureType and GetFeature type of requests. Since such a kind of web feature service supports reading of data only, it is also known as a 'read-only' server.

XLink WFS: In addition to supporting all the operations of a basic web service, an XLink WFS also provides the GetGmlObject operation.

Transaction WFS: A transactional WFS supports all the operations of a basic web server and in addition, it also implements the Transaction operation. A transactional WFS may, optionally, support the GetGmlObject operation as well.

Data

The Geographic Markup Language (GML) is used for encoding the information that passed between a client and a server. Thus, it is required that both the client and the server support and understand GML.

Comparison of Web Feature Service to Web Map Service

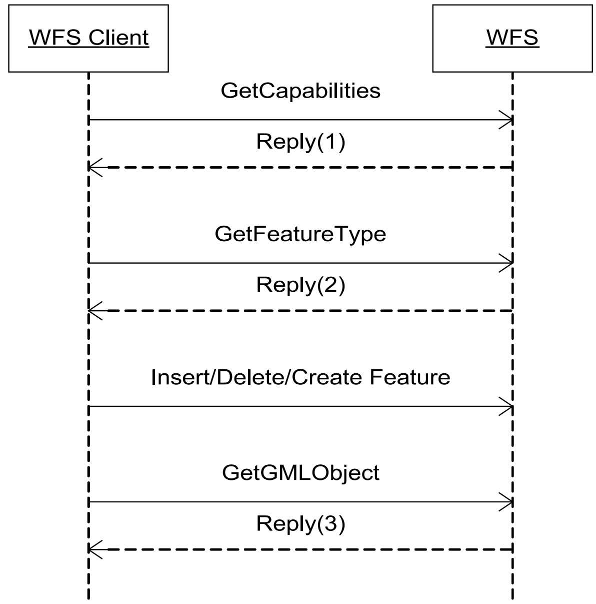
Although WFS is closely related to its cousin, the Web Map Service (WMS), there are significant differences. With WMS, the machine that requests some other machine for information gets a completely rendered map in return for its request. The requesting machine does not get raw data. It gets a readymade map which it merely displays. In contrast, in WFS, the requesting machine gets the raw data for the geographic feature it had requested. This means that geographic co-ordinate data such as line, point and polygon features are returned. It is the prerogative of the requesting machine to use the returned information in whatever manner it wants. The information is returned in the form of GML (Geographic Markup Language), the geographic cousin of XML. Hence, WFS is the means to get feature-level geospatial data and use this data to render a map. The term Feature-level geospatial data is meant as geographic coordinate data such as points, lines and polygon features. Hence, it is understood that a WFS request could potentially involve a large amount of data transmission from a server to a client. Although the client is required to do more work in terms of making use of the geospatial data, WFS offers the client more flexibility. For instance, a client may save the returned features to a usable data file. In contrast, a requesting WMS client gets the rendered map back with no opportunity to use the data in any other way. That is, it may or not display the rendered map. However, since it does not have the raw data associated with the map, it cannot store the data. Though WFS provides more freedom to a client to make use of geospatial data, the work of the client is increased as it must now make appropriate use of that data, without which the data is just "data" and not useful information. Therefore, both WMS and WFS are beneficial in their own way, depending on the need of the client.

Thus, a WFS request consists of a description of the data manipulation that is to be applied to one or more features. The client composes this request and sends it to the server over HTTP. The server (called the web feature server) reads the request and executes it.

Requirements of a Web Feature Server

There are some important requirements that a server must satisfy before it can claim to be a Web feature server.

1. The interfaces must be defined in XML.
2. The server must be able to encode the geographic information using the Geographic Markup Language (GML).
3. The client should not be required to have knowledge of the data store used to store the geographic features. The client's only view of the data should be through the WFS interface.
4. The predicate or filter language must be defined in XML and be derived for the Common Query Language (CQL) as defined in the OpenGIS Catalogue Interface Implementation Specification [2].
5. A Web Feature Service must also be able to describe the structure of each feature type defined by it.
6. It must be possible for a client to specify which feature properties to fetch and to do so using spatial and non-spatial constraints.



Web Feature Service (WFS), Figure 3 Sequence of events

Sequence of Events

Interaction between a web feature server and a client generally follows a sequence, as illustrated in Fig. 3.

The sequence works as follows:

1. A client application would make a request for a capabilities document from the WFS. As the name suggests, this document gives a description of all the operations supported by the WFS as well as the feature types that it can service.
2. Next the client may make a request to the WFS for the definition of one or more of the feature types that the WFS can service. This step is optional and can be safely omitted by the client if the parameters needed to be passed in a request are already known by the client.
3. Based on the definition of the feature type(s), the client generates a request.
4. Once the request reaches the web server, it invokes the WFS to read and service the request.
5. The WFS processes the request. On completion of the request, the WFS will send a status report back to the client. It will also notify the client if an error occurred during processing.

Key Applications

The utility of Web Feature Service is not as visible as its illustrious cousin, the Web Map Service (WMS). This is because web feature service provides interfaces for querying and modifying geospatial information. Data is returned to the requesting client in the form of GML. The visibility and use of this data is realized only after it is rendered in the form of maps. However, it is clear that WFS provides the raw material (data) for the finished product (the rendered map). The tremendous utility of web feature service is that a requestor of information can choose to use the raw data in any number of ways. For instance, the client may store the data or display the data.

Future Directions

As Web Feature Services is a protocol, it is under constant review and changes. This is especially so because it is a protocol that deals with web services in the upcoming area of geospatial services. The latest specification as well as the ones before it can be found at the Open GeoSpatial Consortium (OGC) website (http://www.openspatial.org).

Cross References

- ► Geography Markup Language (GML)
- ► Web Feature Service (WFS) and Web Map Service (WMS)
- ► Web Services, Geospatial

Recommended Reading

1. OpenGeoSpatial Consortium (OGC) specification on Web Feature Service (http://www.opengeospatial.org/standards/wfs): According to the OGC website, (http://www.opengeospatial.org), OGC is a non-profit, international, voluntary consensus standards organization that is leading the development of standards for geospatial and location based services. It provides technical documents that detail agreed-upon interfaces or encodings which when implemented by software developers will ensure that the resulting components work together coherently
2. Web Mapping Illustrated by Tyler Mitchell (O'Reilly Series, 2005). This book explains how to find, collect, understand, use, and share mapping data, both over the traditional Web and using OGC-standard services like WFS and WMS
3. Beginning MapServer: Open Source GIS Development by Bill Kropla (Apress Publications, 2005). This book shows the use of OGC-standard services in MapServer, one of the popular development platforms for integrating mapping technology into Internet applications
4. http://www.en.wikipedia.org: This collaborative web-based, free content encyclopedia project is a good resource to see information on WFS brought together. It provides additional links for further investigation

Web Feature Service (WFS) and Web Map Service (WMS)

CHRISTOPHER D. MICHAELIS, DANIEL P. AMES
Department of Geosciences, Geospatial Software Lab, Idaho State University, Pocatello, ID, USA

Synonyms

Internet mapping service; MapServer;
ArcIMS; ArcExplorer;
Web processing service; WFS; WMS

Definition

Open Geospatial Consortium (OGC) standards for the exchange of geospatial data form a consistent foundation for the development of geographic information systems (GIS) software. The Open Geospatial Consortium (http://www.opengeospatial.org/) is a leading body in standards development in GIS, consisting of members across diverse scientific backgrounds [1]. The most widely adopted and popular of these is the Web Map Service (WMS), a specification which outlines communication mechanisms allowing disjoint software products to request and provide preassembled map imagery ("compiled" map images, which may contain both vector and raster data) to a requesting client. A similar service, Web Feature Service (WFS), allows clients to request vector data which is returned in a pure form (preserving the vector nature of the dataset). Another service which is raising in popularity though not yet a formal specification is the Web Processing Service (WPS) proposal. This illustrates a mechanism by which a client may request geoprocessing to be performed on a server, in a similar fashion to that available in ESRI™ ArcIMS software which also allows clients to perform GIS processing and analysis via the Internet. While ArcIMS protocols to this end are not published, OGC specifications which outline communications protocols aim to be open.

Software (whether client-side or server-side) implementing these key specifications gains the advantage of being interoperable with other geospatial software. An excellent example of this is MapServer, produced by the University of Minnesota, which acts as a server for both WMS and WFS communications mechanisms [3]. WMS and WFS servers may be high-end datacenter servers or even a personal workstation computer. Data exposed by MapServer through these specifications may be consumed by any client software implementation which supports the specifications; e. g., MapWindow GIS, a client-side application, or the Refractions Research web-based client tool (http://mapserver.refractions.net/phpwms/phpwms-cvs/). Servers may even consume data sources from each other, building maps upon resources spread throughout the internet. MapServer again serves as an example: in addition to providing WMS and WFS data, it can also consume WFS and WMS data from other servers to render maps for an end user.

Historical Background

Web-based geographic information system (GIS) tools are becoming very common for many basic mapping and data visualization tasks (e. g., Google^{TM} Maps, MapQuest^{TM}, and ArcIMS). However, desktop or client-side GIS tools are likely to continue to be heavily used by government, academic, and commercial users requiring a high degree of custom data management, modeling, and analysis functionality. The primary standards issuing entity for the GIS community, the Open Geospatial Consortium (OGC), has released many specifications which are not necessarily web-specific, yet the most widely adopted of its standards are specifically for web based GIS tools. Because of this, it is likely that most GIS users and tool developers (specifically those developing tools for the desktop GIS environment) are unaware of OGC standards and specifications and their potential implications. Specific OGC standards such as Web Map Service (WMS) and Web Feature Service (WFS) have significant potential for normalizing, and hence improving, the manner in which data is shared across the Internet. Most day to day users may be concerned with how to use data provided by these services in their client-side GIS applications, but many may also be interested in using this data in their own server-based tools for providing maps.

The stated mission of the OGC is "to lead the global development, promotion and harmonization of open standards and architectures that enable the integration of geospatial data and services into user applications [2]." This is accomplished through the authoring of specifications which are created by "structured committee programs and a consensus process" by which participants in a wide range of scientific disciplines may contribute to the specifications. A specific goal of the OGC is to address the problem of data sharing – a problem that includes both interoperability and communication – and ultimately arrive at "a world in which everyone benefits from the use of geospatial information and supporting technology". OGC has attempted to address some of the fundamental problems of data sharing by specifying a common format for wide-scale understandability in distribution, for facilitating geoprocessing, and for interoperability between disjoint software products. It is of note that the OGC produces standards with respect to data transfer and communications; but not in metadata areas. The FGDC and ISO standards are the main issuing entities for metadata standards.

Scientific Fundamentals

A tightly defined XML (eXtensible Markup Language) schema may easily keep servers and clients interoperating successfully despite being written by varied authors for various platforms. Most OGC specifications make use of XML schemas in this manner. The first such example is Web Map Service (WMS), with which a client may request a map image from a server. This is then returned in a standard imaging format (e. g., JPEG). The request is authored using XML such that the server will understand exactly how to parse and understand the request. The image is returned without any geographic location information (e. g., a "world file"), since this service is primarily used for end-user interfaces and not mosaic operations or similar operations requiring geographic location information. Frequently, server implementations will return exactly the spatial extents requested, making the returning of this particular metadata less important.

In WMS, the key communications are performed using XML documents which list the available data on a server (the GetCapabilities request), which provide details on a given dataset (the DescribeLayer request), or which request a map image as described above (the GetMap request).

The Web Feature Service (WFS) uses XML more extensively. Similarly to WMS, operations to list data on a server (GetCapabilities) and to describe a dataset (DescribeLayer) are used. Rather than returning an image from a GetMap request, WFS will provide vector data, without first assembling it into a standard image format (which tends to lose data quality and precision). The word "feature" in this service is referring to a vector object such as a point, line, or polygon, and the word carries some history of its own. A client may request vector data from a server in a manner similar to that used in WMS (a tightly defined XML schema is used). The server responds with the requested vector information encoded in Geographic Markup Language, a separate standard which defines a common and open format for communicating geographic and geometric data. Because WFS uses GML, which is also based on XML, this causes a greater quantity of XML text to be transferred. Some advantages to this are that XML is designed to be easily parsed by machine yet also easily readable by humans – thus making debugging easier and helping the user to understand what is taking place. A major disadvantage, however, is the overhead associated with transmitting XML tags and all data in an essentially text-based format. Binary data transmission could cut the total number of bytes to be transmitted nearly in half – a staggering amount of overhead. While this overhead is huge, it is often disregarded due to high internet connection speeds and due to the ease of working with XML data. WFS is not as heavily adopted as WMS, potentially due to the greater overhead and computational power required to use it, but is nonetheless used widely across the internet.

An example of when WFS may be appropriate rather than WMS may be if geospatial operations will need to be performed on the data (e. g., for a calculation or for a model), rather than simply using the data for end user visualization. A related service, Web Processing Service, has been proposed but not yet formally accepted as an Open Geospatial Consortium specification. This proposal outlines a mechanism by which a server may perform geoprocessing operations as requested by a client, using XML communication mechanisms very similar to those observed with WFS and WMS. Though the proposal has not been adopted widely, some investigation into the feasibility of WPS has been done and shows great promise. Using the system outlined by the proposal, a client may encode input data and transmit it to a server, requesting a given operation to be performed. Processing may take place on the server, with output data being returned back to the client upon completion. GML is again used for vector data, and binary formats or base-64 encoded data are often used for raster data.

Key Applications

The best known software package implementing WFS and WMS is MapServer. This package acts as both a client and server for WFS and WMS data, meaning that it can serve data in these formats as well as consume this data from other servers. Other applications that can consume WFS and WMS data include OpenEV, ArcExplorer, MapWindow GIS and uDig, among others.

Future Directions

WFS and WMS have already shown great potential for providing data using a common communications mechanism across the internet. Server tools and end-user applications will undoubtedly continue to adopt these standards, allowing for further data sharing. WPS, though it hasn't been widely used or adopted, shows great potential as applications developers begin to adopt it and make greater use of the proposed specification.

Cross References

► Web Services

Recommended Reading

1. Open Geospatial Consortium, Inc. Filter Encoding Implementation Specification, OGC Website: http://www.opengeospatial.org/ (2001)
2. Open Geospatial Consortium, Inc. Vision and Mission, OGC Website: http://www.opengeospatial.org/about/?page=vision (2006)
3. University of Minnesota. MapServer, MapServer Website: http://mapserver.gis.umn.edu/ (2006)

Web GIS

► Internet GIS

Web Graphics Standard

► Scalable Vector Graphics (SVG)

Web Map Server

► University of Minnesota (UMN) Map Server

Web Map Service

► deegree Free Software
► OGC's Open Standards for Geospatial Interoperability
► Web Feature Service (WFS)

Web Mapping and Web Cartography

ANDREAS NEUMANN
Institute of Cartography, ETH Zurich, Zurich, Switzerland

Synonyms

Electronic atlases; Maps on internet; Mobile maps; SVG; Web technologies; CGI; Web server; Web application server; Web mapping server; WMS

Definition

Web mapping is the process of designing, implementing, generating and delivering maps on the World Wide Web. While web mapping primarily deals with technological issues, *web cartography* additionally studies theoretic aspects: the use of web maps, the evaluation and optimization of techniques and workflows, the usability of web maps, social aspects, and more. *Web GIS* or *Internet GIS* is related to web mapping but with an emphasis on analysis, processing of project specific geodata as well as exploratory aspects. Often the terms web GIS and web mapping are used synonymously, even if they don't mean the same. In fact the boundary between web maps and web GIS is blurry. Web maps are often a presentation media in web GIS and web maps are increasingly gaining analytical capabilities. A special case of web maps are *mobile maps*, displayed on mobile computing devices, such as mobile phones, smart phones, PDAs, GPS and other devices. If the maps on these devices are displayed by a mobile web browser or web user agent, they can be regarded as mobile

web maps. If the mobile web maps also display context and location sensitive information, such as points of interest, the term *location based services* is frequently used.
The use of the web as a dissemination medium for maps can be regarded as a major advancement in cartography and opens many new opportunities, such as realtime maps, cheaper dissemination, more frequent and cheaper updates of data and software, personalized map content, distributed data sources and sharing of geographic information. It also implicates many challenges due to technical restrictions (low display resolution and limited bandwidth, in particular with small devices), copyright and security issues, reliability issues and technical complexity. While the first web maps were primarily static, due to technical restrictions, today's web maps can be fully interactive and integrate multiple media. This means that both web mapping and web cartography also have to deal with interactivity, usability and multimedia issues.

Historical Background

Naturally, the history of web maps is closely tied to the history and technological advancements of the web. For a more complete list of web mapping and web technology related events, see [13]. Until the advent of the WWW, invented at CERN in 1989 and released to the public in 1993, the data transfered over the internet was primarily based on text or proprietary data formats. It is not clearly documented when the first map was published on the web, but shortly after HTML became a standard, in June 1993, Xerox Parc already implemented the first interactive map server on the web. This CGI map server implementation written in Perl allowed the user to specify a map extent, choose map layers, a map projection and some styling parameters. The response was a HTML file and an embedded gif raster image representing the map. It can be regarded as the predecessor of today's standardized WMS (Web Map Service). In 1994, the National Atlas of Canada went online as the first major online atlas worldwide. Later that year, the first version of the popular Netscape browser was released.
In March 1995, Sun made the first version of Java available, and was introduced in the main web browsers soon after. Java Applets are a huge step forward regarding interactive, platform independent applications on the web. From the beginning in 1995 until now, Java applets were used to deliver interactive map content on the web. In December 1995, Brendan Eich (Netscape) introduced the first version of Javascript which enabled first interactive websites, including simple interactivity in raster based maps.
Early in 1996, Mapquest, one of the first popular address matching and routing services went online. Results were displayed using web maps and textual descriptions. In the middle of 1996, Multimap, a UK based web mapping service went online. It is now one of the UKs most popular websites. In late 1996, Geomedia WebMap 1.0 was published, one of the first commercial web map server. At about the same time, Macromedia released the Flash Player 1.0, after buying the technology from Futurewave. Gradually, Flash grew popular for multimedia and animation content on the web, but it took a couple years (until about 2000), until Flash was suitable for more serious applications with a mature scripting language available. In 1997, USGS received the mandate to coordinate and create the Online National Atlas of the United States.
In the middle of 1997, UMN MapServer 1.0 was released. It grew into the most popular open source mapping server and is used by thousands of web mapping sites today. Already in December 1997, HTML 4.0 was published, the first version that allowed styling with CSS and absolute positioning using pixels. Scripting at this time was still highly proprietary. Web authors had to write different script code for the major webbrowsers: Netscape Navigator and Microsoft Internet Explorer. This period is also well-known as the *browser wars*, a time in which browser vendors introduced many proprietary features in the HTML language to set themselves apart from the competition, also a time where users and web developers suffered a lot from the incompatibilities between the different web browsers.
In June 1998, a joint effort from USGS, Microsoft and HP started the popular US Terraserver project, an OGC compatible WMS server, serving aerial images and USGS topo maps. Shortly after, ESRI entered the web map server market with its MapObjects Internet Map Server, a project that was later replaced by the ArcIMS server (2000). In September 2001, W3C released the SVG 1.0 specification, a XML based markup language for the integration of vector graphics, raster graphics and text, also supporting animation, multimedia, scripting and interactivity, and internationalization. SVG opened many opportunities for interactive web maps but it took a while until SVG was properly supported in web browsers. The Tirol Atlas, released in 2001 shortly after SVG 1.0 becoming a recommendation, is the first major online atlas extensively using SVG technology. It is still one of the few atlases offering superior interactivity.
In January 2003, the SVG mobile profiles SVG tiny and SVG basic were released, enabling location based services and mobile maps. In the same year, NASA released *WorldWind*, one of the first interactive virtual earth applications. This software requires a special virtual globe software and does not run in web browsers, but it offers open, OGC compatible interfaces and data is loaded across the

internet. In early 2005, Google released *Google Maps* a web mapping application developed on top of Dynamic HTML, ECMAScript and XMLHttpRequests (also known as Ajax). The fact that Google opened its API for third party developers for free re-use made it quickly highly popular, triggering thousands of Google Maps derived web mapping applications. Technically, Google Maps is based on quad-tree raster tiles of aerial images and road maps and a geospatial search engine. Later, in the same year, Google released *Google Earth*, a virtual earth application bought from Keyhole. KML, a XML based markup language, allowed users to add their own 3D geodata content. In November 2005, Mozilla released Firefox 1.5, the first version with native SVG support, followed in 2006 by Opera 9, which currently offers the best native SVG support. Apple and KDE will release native SVG support in Safari and Konqueror during 2007. In August 2006, SVG Tiny 1.2 went to W3C candidate recommendation, with improved multimedia support and better support for building rich client internet applications.

Scientific Fundamentals

The advent of web mapping can be regarded as a major new trend in cartography. Previously, cartography was restricted to a few companies, institutes and mapping agencies, requiring expensive and complex hardware and software as well as skilled cartographers and geomatics engineers. With web mapping, freely available mapping technologies and geodata potentially allow every skilled person to produce web maps, with expensive geodata and technical complexity (data harmonization, missing standards) being two of the remaining barriers preventing web mapping from fully going mainstream. The cheap and easy transfer of geodata across the internet allows the integration of distributed data sources, opening opportunities that go beyond the possibilities of disjoint data storage. Everyone with minimal knowhow and infrastructure can become a geodata provider. These facts can be regarded both as an advantage and a disadvantage. While it allows everyone to produce maps and considerably enlarges the audience, it also puts geodata in the hands of untrained people who potentially violate cartographic and geographic principles and introduce flaws during the preparation, analysis and presentation of geographic and cartographic data.

Advantages of Web Maps

Web maps can easily *deliver up to date information*. If maps are generated automatically from databases, they can display information in almost realtime. They don't need to be printed, mastered and distributed. Imagine a map displaying election results as soon as the election results become available, or a map displaying the traffic situation near realtime by using traffic data collected by sensor networks. Because web maps distribute both logic and data with each request or loading, product updates can happen every time the web user reloads the application. In traditional cartography, a map update caused serious efforts triggering a reprint or remastering as well as a redistribution of the media. With web maps, data and product updates can happen with considerably less efforts and costs, in a much shorter time span and with more dense update intervals. *Web server hardware is cheaply available* and many open source tools exist for producing web maps. If web maps are implemented based on open standards, the underlying operating system and browser do not matter. If properly implemented, *web maps work across browsers and operating systems*. It is also easy to *integrate multimedia* in and with web maps. Current web browsers support the playback of video, audio and animation. *Web maps also support hyperlinking and act as an information hub*. By using vector geometries or sensitive areas in a web map, any portion of the map can link to other web pages or web services. As an example, a city map can link to the corresponding timetables of the public transport system at every bus or train station. *Web maps can combine distributed data sources*. Using open standards and documented APIs one can integrate different data sources, if the projection system, map scale and data quality match. The use of centralized data sources removes the burden for individual organizations to maintain copies of the same data sets. The down side is that one has to rely and trust the external data sources.

By using user profiles, personal filters and personal styling and symbolization, *users can personalize, configure and design their own maps*. Accessibility issues can be treated in the same way. If users can store their favorite colors, symbols and patterns they can avoid color combinations and map symbolizations they can't easily distinguish, e. g. due to color blindness. *Web maps also enable collaborative mapping*. Similar to the Wikipedia project, web mapping technologies, such as DHTML/Ajax, SVG, Java, Adobe Flash, etc. enable distributed data acquisition and collaborative efforts. Examples for such projects are the OpenStreetMap project [8] or the Google Earth community [2]. As with other open projects, quality assurance is essential.

Problems with Web Maps

The realibility of the internet and web server infrastructure is not yet good enough. Especially if a web map relies on external, distributed data sources, the original author cannot guarantee the availability of the information. They usually require a relatively high bandwidth. Despite the increasing availability of free and commercial tools to create web mapping and web GIS applications, *web maps are still complex to develop.* Many technologies, modules, services and data sources have to be mastered and integrated. Compared to the development of standalone applications with integrated development tools, the development and debugging environments of a conglomerate of different web technologies is still awkward and uncomfortable.

Geodata is often expensive. Unlike in the USA, where geodata collected by governmental institutions is usually available for free or cheap, geodata is usually very expensive in Europe or other parts of the world. This is a serious barrier for many low budget web mapping projects in Europe and decreases quality of web maps due to limited access to high quality geodata. There are also open *copyright and privacy issues.* Many people are still reluctant to publish geodata, especially in the light that geodata is expensive in some parts of the world. They fear copyright infringements of other people using their data without proper requests for permission. Digital rights managements does not work reliably and is usually cracked in a short time by hackers. With detailed information available and the combination of distributed data sources, it is possible to find out and combine a lot of private and personal information of individual persons. High resolution aerial and satellite images of private properties and estates are now easily accessible throughout the world to anyone.

Finally, as with any other screen based map, there is the problem of *limited screen space*. This is in particular a problem for mobile web maps and location based services where maps have to be displayed on very small screens with resolutions as low as 100x100 pixels. Hopefully, technological advances will help to overcome these limitations.

Types and Properties of Web Maps

One of the first classifications of electronic maps and atlases was carried out by [11]. They distinguished *view-only atlases*, *atlases that generate maps on demand* and *analytical atlases based on GIS capabilities*. [9] used the terms *view-only atlases*, *interactive atlases* and *analytical atlases* for his classification of electronic atlases. A first classification of web maps was made by [4]. He distinguished between static and dynamic web maps and further differentiated between interactive and view only web maps. However, today in the light of an increased number of different web map types, this classification needs some revision. There are additional possibilities regarding distributed data sources, collaborative maps, personalized maps, and many more. It is impossible to create a complete classification covering all types of web maps. However, various properties of a web map can be defined. Figure 1 lists potential properties of web maps with each property as a value-pair. Every product can be allocated to either property of the value pair or in between on a scale between 0 and 1. Obviously, this list of properties is not complete and more properties can be added as web maps gain capabilities.

Following is a list of web map types with a short description of their properties and characteristics. Obviously, many maps can be allocated to more than one web map type:

Static web maps are view only with no animation and interactivity. They are only created once, often manually and infrequently updated. Typical graphics formats for static web maps are png, jpeg or gif for raster files, svg, pdf or swf for vector files. Often, these maps are scanned paper maps and had not been designed as screen maps. Paper maps have a much higher resolution and information density and might be illegible when displayed on screens at the wrong resolution.

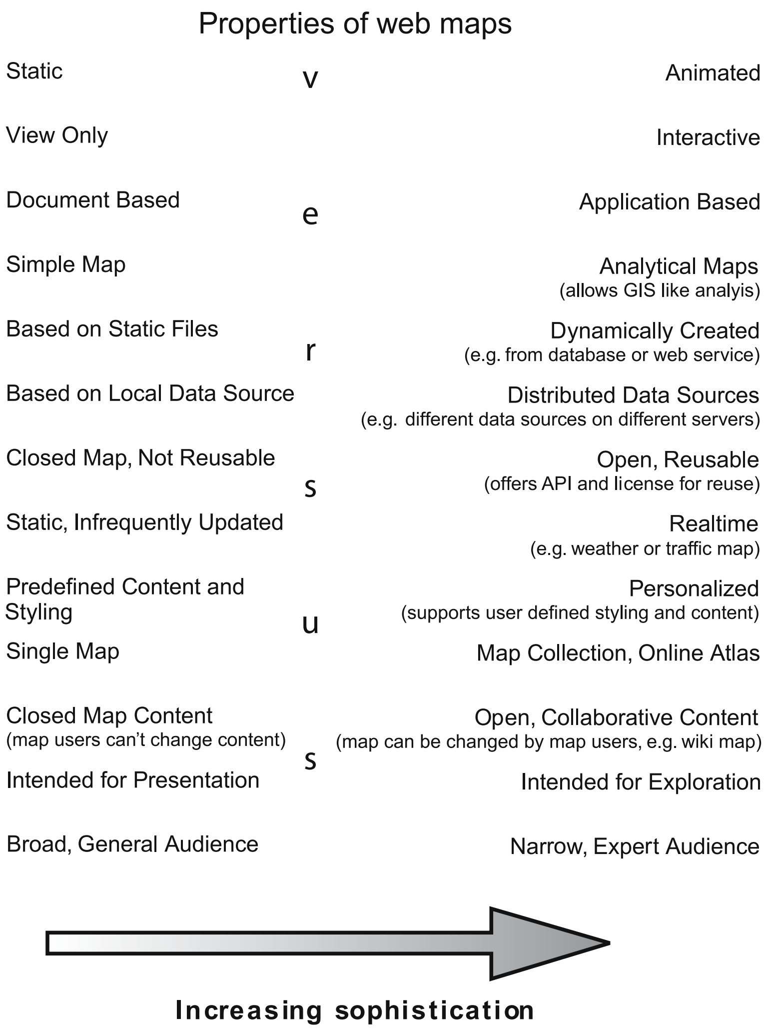
Dynamically created web maps are created on demand each time the user reloads the webpages, often from dynamic data sources, such as databases. The webserver generates the map using a web map server or a self written software.

Distributed web maps are created from distributed data sources. The WMS protocol offers a standardized method to access maps on other servers. WMS servers can collect these different sources, reproject the map layers, if necessary, and send them back as a combined image containing all requested map layers. One server may offer a topographic base map, while other servers may offer thematic layers.

Animated web maps show changes in the map over time by animating one of the graphical or temporal variables. Various data and multimedia formats and technologies allow the display of animated web maps: SVG, Adobe Flash, Java, Quicktime, etc., also with varying degrees of interaction. Examples for animated web maps are weather maps, maps displaying dynamic natural or other phenomena (such as water currents, wind patterns, traffic flow, trade flow, communication patterns, etc.).

Realtime web maps show the situation of a phenomena in close to realtime (only a few seconds or minutes delay). Data is collected from sensors, sent across the internet to a central server, and the maps are generated or updated at regular intervals or immediately on demand. Examples are weather maps, traffic maps or vehicle monitoring systems.

Personalized web maps allow the map user to apply his own data filtering, selective content and the application of personal styling and map symbolization. The OGC consortium provides the SLD standard (Styled Layer Descrip-



Web Mapping and Web Cartography, Figure 1
Potential properties of web maps

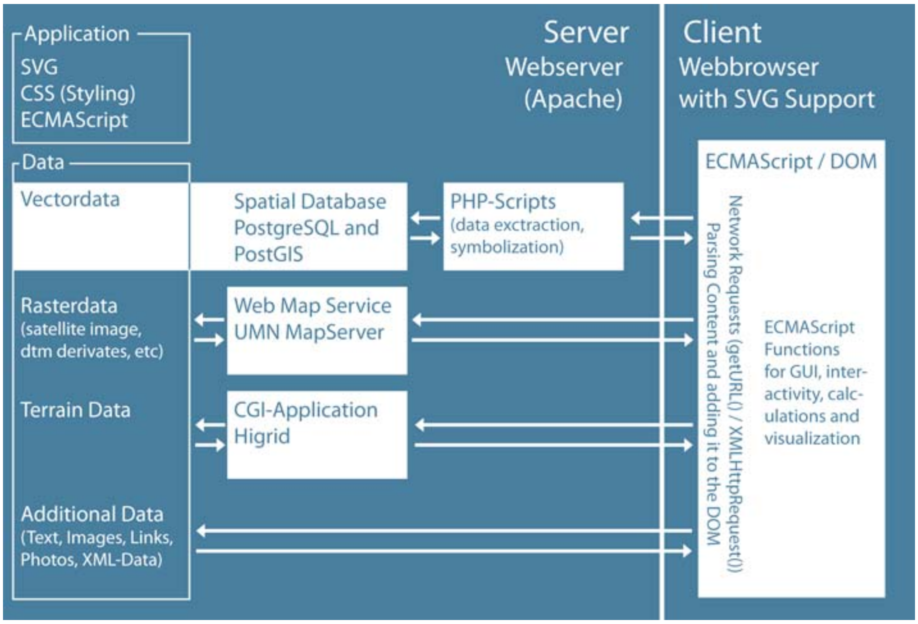
tion) that may be sent to a WMS server for the application of individual styles. This implies that the content and data structure of the remote WMS server is properly documented.

Open, reusable web maps are usually more complex web mapping systems that offer APIs for reuse in other people's web pages and products. An example for such a system is Google Maps with the Google Maps API. Ideally, such APIs would be compatible with the standards promoted by the Open Geospatial and W3C Consortium, but unfortunately, in reality they are often proprietary.

Interactive web maps help to compensate for the disadvantages of screen and web maps (limited screen space, bad resolution, limited color ranges, etc.). Interactivity helps to explore maps, change map parameters, navigate and interact with the map, reveal additional information, link to other resources, and much more. Technically, it is achieved through the combination of events, scripting and DOM manipulations.

Analytic web maps offer GIS analysis, either with geodata provided, or with geodata uploaded by the map user. As already mentioned, the borderline between analytic web maps and web GIS is blurry. Often, parts of the analysis is carried out by a server side GIS and the client displays the result of the analysis. As web clients gain more and more capabilities, this task sharing may gradually shift.

Collaborative web maps are still new, immature and complex to implement, but show a lot of potential. The idea is that, like in the Wikipedia project, various, distributed, people collaborate to create and improve maps on the web. Technically, an application allowing simultaneous editing across the web would have to ensure that geometric features being edited by one person are locked, so that they can't be edited by other persons at the same time. Also, a minimal quality check should be made, before data goes public. Some projects working on collaborative maps are OpenStreetMap, Google Earth and WikiMapia.



Web Mapping and Web Cartography, Figure 2 Example of a web mapping architecture. Source: [1, p. 4]

Web Mapping Architectures

The potential number of technologies to implement web mapping projects is almost infinite. Any programming environment, programming language and server-side framework can be used to implement web mapping projects. In any case, both server and client side technologies have to be used. Following is a list of potential and popular server and client side technologies utilized for web mapping. Figure 2 shows one potential web mapping architecture that was used for a web map on *Slope Stability on Nisyros Island (Greece)*. It includes the most important components of a web mapping architecture, but obviously some modules could be replaced by other software and some modules aren't needed for certain types of web maps.

Server Side Technologies

The *webserver* is responsible for handling http requests by web browsers and other user agents. In the simplest case they serve static files, such as HTML pages or static image files. Web servers also handle authentication, content negotiation, server side includes, URL rewriting and forward requests to dynamic resources, such as CGI applications or serverside scripting languages. The functionality of a webserver can usually be enhanced using modules or extensions. The most popular web server is Apache, followed by Microsoft Internet Information Server, Netscape and others.

CGI (common gateway interface) applications are executables running on the webserver under the environment and user permissions of the webserver user. They may be written in any programming language (compiled) or scripting language (e. g. perl). A CGI application implements the common gateway interface protocol, processes the information sent by the client, does whatever the application should do and sends the result back in a web-readable form to the client. As an example, a web browser may send a request to a CGI application for getting a map with a certain map extent, styling and map layer combination. The result is an image format, e. g. jpeg, png or SVG. For performance enhancements one can also install CGI applications as FastCGI. This loads the application after the web server is started and keeps the application in memory, eliminating the need to spawn a separate process each time a request is being made. Alternatively, one can use *scripting languages* built into the webserver as a module, such as PHP, Perl, Python, ASP, Ruby, etc. If built into the web server as a module, the scripting engine is already loaded and doesn't have to be loaded each time a request is being made.

Web application servers are middleware which connect various software components with the web server and a programming language. As an example, a web application server can enable the communication between the API of a GIS and the webserver, a spatial database or other proprietary applications. Typical web application servers are written in Java, C, C++, C# or other scripting languages. Web application servers are also useful when developing complex realtime web mapping applications or Web GIS.

Spatial databases are usually object relational databases enhanced with geographic data types, methods and properties. They are necessary whenever a web mapping application has to deal with frequently changing dynamic data or with huge amounts of geographic data. Spatial databases allow spatial queries, sub selects, reprojections, geometry manipulations and offer various import and export formats. A popular example for an open source spatial database is PostGIS. MySQL also implements some spatial features, although not as mature as PostGIS. Commercial alternatives are Oracle Spatial or spatial extensions of Microsoft SQL Server and IBM DB2. The OGC Simple Features for SQL Specification is a standard geometry data model and operator set for spatial databases. Most spatial databases implement this OGC standard.

WMS server are specialized web mapping servers implemented as a CGI application, Java Servlet or other web application server. They either work as a standalone web server or in collaboration with existing web servers or web application servers (the general case). WMS Server can generate maps on request, using parameters, such as map layer order, styling/symbolization, map extent, data format, projection, etc. The OGC Consortium defined the WMS standard including the map requests and allowed return data formats. Typical image formats for the map result are png, jpeg, gif or SVG. An open source WMS Server is the UMN Mapserver. Commercial alternatives exist from most commercial GIS vendors, such as ESRI ArcIMS, Intergraph Geomedia WebMap and others.

Client Side Technologies

In the simplest setup, only a *web browser* is required. All modern web browsers support the display of HTML and raster images (jpeg, png and gif format). Some solutions require additional plugins (see below). *ECMAScript* is the standardized version of JavaScript. It is necessary to implement client side interaction, refactoring of the DOM of a webpage and for doing network requests. ECMAScript is currently part of any modern web browser. Various *events* are necessary to implement interactive client side maps. Events can trigger script execution or SMIL operations. One distinguishes between mouse events, keyboard events, state events, mutation events, SMIL animation events, UI events and SVG specific events. Network requests are necessary to load additional data and content into a web page. Most modern browsers provide the XMLHttpRequest object which allows get and post http requests and provides some feedback on the data loading state. The data received can be processed by ECMAScript and can be included into the current DOM tree of the web page or web map. SVG user agents alternatively provide the getURL() and postURL() methods for network requests. These network requests are also known under the term Ajax.

The *Document Object Model* provides a language independent API for the manipulation of the document tree of the webpage. It exposes properties of the individual nodes of the document tree, allows to insert new nodes, delete nodes, reorder nodes and change existing nodes. DOM support is included in any modern web browser. DOM support together with scripting is also known as DHTML or Dynamic HTML. Google Maps, Microsoft Local and many other web mapping sites use a combination of DHTML, Ajax, SVG and VML. SVG is the abbreviation of *Scalable Vector Graphics* and integrates vector graphics, raster graphics and text. SVG also supports animation, internationalization, interactivity, scripting and XML based extension mechanisms. SVG is a huge step forward when it comes to delivering high quality, interactive maps. At the time of writing (2007-01), SVG is natively supported in Mozilla Firefox (version 1.5 or later), Opera (version 9 or later) and the developer version of Safari Webkit. Internet Explorer users still need the Adobe SVG viewer plugin provided by Adobe. [7] discusses the use of SVG for interactive web mapping.

Since 1995, *Java applets* can be used in web browsers. Some browsers still provide old versions of the Java virtual machine. An alternative is the use of the Sun Java Plugin. Java is a full featured programming language that can be used to create very sophisticated and interactive web maps. The Java2D and Java3D libraries provide 2d and 3d vector graphics support. The creation of Java based web maps requires a lot of programming know how. [3] discusses the use of Java applets for the presentation of interactive choroplethe maps and cartograms.

Web browser plugins are a way to extend the capabilities of a web browser. While this works fine in theory, browser plugins create a lot of problems, because plugins often cannot interact properly with the rest of the browser and cannot be blended easily with other contents of a web page. A popular plugin is *Adobe Acrobat* for viewing PDF files. PDF files may display map layers (they can be interactively turned on and off) and contain interactivity, links and scripting. Another popular plugin is *Adobe Flash* (former Macromedia), providing vector graphics, animation and multimedia support. It allows the distribution of sophisticated interactive maps, as with Java and SVG. It also features a programming language (ActionScript) which is similar to ECMAScript and supports Audio and Video. A gallery of highly interactive Adobe Flash based maps can be seen at [6]. *Apple Quicktime* adds support for additional image formats, video, audio and Quicktime VR (Panorama Images). The *Adobe SVG viewer* provides SVG 1.0 support for web browsers with no native SVG support. The *Sun Java plugin* provides support for newer and advanced Java Features.

Key Applications

Address Matching, Routing

A classic use case of web maps are the display of the results of address searches and route finding operations. Typically, the searches and route finding algorithms are executed on the web server and the resulting map sent back to the client, with the search result highlighted. The results of a route finding operation may be animated and interactivity can be used for the display of additional information (such as points of interest, accompanying text at junctions, etc.).

Realtime Maps

Realtime web maps fully exploit the strengths of web maps. Sensors send their data to a central server repository which is used to generate realtime (or near realtime) maps on demand. Examples are weather maps, traffic maps, maps for fleet management, etc. Once a map is loaded it can update the dynamic information on demand, without having to reload the application or base map. This is technically implemented using network requests (XMLHttpRequests or sockets).

Location Based Services

Location based services (LBS) need maps to display points of interest. Mobile web maps are displayed by relying on mobile browsers, SVG, Flash or Java based user agents. Mobile web maps need to be specially designed, taking into account the small screen space and low resolution of mobile displays. Most of the time, maps for location based services are generated automatically from spatial databases.

Urban and Regional Planning

Web maps can support the urban and regional planning process by informing the concerned public and decision makers through a visualization of planned activities and measures. Depending on the type of the web map, web maps can be used for presentation purposes or also enable a feedback process by letting the public provide feedback using interactive maps. As an example, a web map could present different scenarios and let the public vote and help decide which action should be taken. A more advanced version would allow the public to directly create and manipulate features in a web map to create their own, preferred scenario. Animated maps are especially useful when presenting simulations or visualizing measures that have different effects over time.

Online Atlases

Atlas projects often went through a renaissance when they made a transition to a web based project. In the past, atlas projects often suffered from expensive map production, small circulation and limited audience. Updates were expensive to produce and took a long time until they hit the public. Many atlas projects, after moving to the web, can now reach a wider audience, be produced cheaper, provide a larger number of maps and map types and integrate with and benefit from other web resources. The first atlas which went online is the *Atlas of Canada*, followed by the *US National Atlas* and commercial offerings, such as Microsoft Encarta Online. The *Tirol Atlas* is the first online atlas making substantial use of SVG and open source web mapping technology. Some atlases even ceased their printed editions after going online, sometimes offering printing on demand features from the online edition. Some atlases (primarily from North America) also offer raw data downloads of the underlying geospatial data sources.

E-Learning

Web maps can be included in e-learning systems. They can illustrate geospatial phenomena, provide a training environment of spatial planning or operations and may even be used as test environments, testing the spatial knowledge and abilities of students. Interactivity, multimedia and animation may support the learning process.

Future Directions

The development and use of web mapping applications has just started and the potential of web maps has only been touched. Many opportunities and use cases of web maps are still to be explored. Realtime maps, distributed maps and especially collaborative maps seem to offer many new possibilities that go beyond what was possible with traditional printed maps or interactive maps distributed on offline media. The advances of GIS and client side programming will bring more analytic maps to the web. Additional interesting research areas are accessibility, internationalization (multi-language maps), personalization and customization. The fact that most big web companies (Google, Microsoft, Yahoo, etc.) are investing in web mapping and virtual earth technology and many governmental agencies are promoting and developing web mapping and web GIS solutions emphasize the importance of the technology. Hopefully, the web will keep and extend its open and collaborative nature also to the area of web mapping and will not end up with proprietary and incompatible, competing solutions. Interoperability of existing and upcoming web mapping solutions should therefore be of high priority. The work of the Open Geospatial Consortium is of high relevance in this area.

Cross References

► Distributed Geospatial Computing (DGC)
► Internet GIS
► Mobile Usage and Adaptive Visualization
► Multimedia Atlas Information Systems
► OGC's Open Standards for Geospatial Interoperability
► Scalable Vector Graphics (SVG)
► User Interfaces and Adaptive Maps

Recommended Reading

1. Flüeler, I., Iosifescu-Enescu, I., Neumann, A.: Cartographic SVG Applications as Risk Management Support and Communication Platforms. In: Proceedings of the Workshop "Visualization, Analytics & Spatial Decision Support" at GIScience 2006 in Münster, Germany (2006)
2. Google Inc. (2006): Google Earth, http://earth.google.com/, accessed 2007-01-05
3. Herzog, A.: Developing Cartographic Applets for the Internet. In: Peterson, M. (ed.) Maps and the Internet, pp. 117–130. Elsevier, Kidlington, Oxford, UK (2003)
4. Kraak, M.-J.: Settings and needs for web cartography, In: Kraak, Brown, A (eds.) Web Cartography. p. 3–4, Taylor & Francis, New York (2001)
5. Kraak, M.-J., Brown, A. (eds.) Web Cartography – Developments and prospects, Taylor & Francis, New York, ISBN 0-7484-0869-X (2001)
6. Mauvière, E., Coulomb, I.: Geoclip, Interactive Flash Map Gallery, http://www.geoclip.fr/an/p24_atlas.php, (2006) accessed 2006-12-28
7. Neumann, A., Winter, A.M.: Webmapping with Scalable Vector Graphics (SVG): Delivering the promise of high quality and interactive web maps, In: Peterson, M. (ed.) Maps and the Internet, p. 197–220. Elsevier, Kidlington, Oxford, UK (2003)
8. OpenStreetMap Community, OpenStreetMap, http://www.openstreetmap.org/ (2006) accessed 2007-01-05
9. Ormeling, F. Atlas information systems, In: Proceedings of the 17th International Cartographic Conference of the ICA, Barcelona, vol. 2, p. 2127–2133 (1995)
10. Peterson, M. (ed.) Maps and the Internet, Elsevier, Oxford, 2003, ISBN 0-08-044201-3, 451 pages (2003)
11. Siekierska, E.M., Taylor, F.D.R.: Electronic Mapping and Electronic Atlases: New Cartographic Products for the Information Era – The Electronic Atlas of Canada, CSIM Journal ACSGC, **45**(1), 11–21 (1991)
12. Tyler, M.: Web Mapping Illustrated – Using Open Source GIS Toolkits, O'Reilly, Sebastopol CA (2003) ISBN 0-596-00865-1, 349 pages
13. Wikipedia – Webmapping. History of web mapping, http://en.wikipedia.org/wiki/Web_mapping#History_of_web_mapping (2007) accessed 2007-01-05

Web Mapping Server

► Web Mapping and Web Cartography

Web Processing Service

► Web Feature Service (WFS) and Web Map Service (WMS)

Web Server

► Web Mapping and Web Cartography

Web Services

ILYA ZASLAVSKY
San Diego Supercomputer Center,
University of California-San Diego, San Diego, CA, USA

Synonyms

Simple object access protocol services; Representational state transfer services; Service-oriented architecture building blocks

Definition

Self-contained and modular web applications that interact with and exchange information with other applications, and can be invoked over the network using a standardized internet-based communication interface.

Main Text

Web services were designed to offer interoperability among diverse and heterogeneous applications, and represent the first interoperability initiative backed by both Microsoft .NET and Java EE development communities. The term "web service" typically refers to services that follow the W3C-defined simple object access protocol (SOAP) and web service description language (WSDL) standards. However, broadly speaking, web services stand for any modular independent software components accessed over the network in a platform and language-neutral way. Representational state transfer (REST) is another type of service that relies on lightweight HTTP interfaces and implements HTTP GET, DELETE, POST and PUT methods. Most interface specifications developed within the Open Geospatial Consortium use a flavor of REST services for retrieving data and capability information from standard-compliant spatial servers (e. g., WMS, WFS, WCS server specifications), though recently Open Geospatial Consortium (OGC) efforts have focused on porting OGC web services to SOAP. Geographic information systems (GIS) and location-based web services are being developed by many software vendors

W

(e. g., MapPoint services from Microsoft, ArcWeb services from ESRI, providing a similar set of services that include Address Finder, Place Finder, Route Finder, Map Image, etc.), as well as within several cyberinfrastructure projects.

Cross References

- ► Cyberinfrastructure for Spatial Data Integration
- ► Grid
- ► Spatial Information Mediation
- ► Web Services

Web Services, Geospatial

CLODOVEU A. DAVIS JR.,
LEONARDO LACERDA ALVES
Institute of Informatics, Pontifical Catholic University of Minas Gerais, Belo Horizonte, Brazil

Synonyms

Spatial web services; Services, geographic information; Services, location-based

Definition

Web services are a specific type of network services that use open internet standards, such as connection and communication through the Hypertext Transfer Protocol (HTTP) and Simple Object Access Protocol (SOAP), identification using the Uniform Resource Identifier (URI), contents specification through the Extensible Markup Language (XML) and Geographic Markup Language (GML), and service descriptions expressed by the Web Services Definition Language (WSDL).

While general web services provide interoperability between systems in different domains, geospatial web services introduce geographic reasoning to the services and go a step further by facilitating cross-institutional interchange of geographic data and services over the internet, and by improving the sharing of geographic resources among a variety of data sources. Thus, geospatial web services can reduce redundant efforts when geospatial data is created and disseminated, while contributing to the reuse of code between different applications.

Geospatial web services differ from the regular web services by the presence of geographic data on the input (e. g., a bounding box), or on the output (e. g., a basemap of a city), or even on the type of processing (e. g., verification of whether a street crosses another), or in a combination of them.

Historical Background

The concept of web services derives from the concept of services in computer networks, in which a service corresponds to a function or to an interface between two abstraction layers of the network design. In that context, a lower-level layer offers services to a higher-level layer, and the higher-level layer is not affected if some property of the lower-level layer is modified.

The idea of isolating specific technical detail through the use of different abstraction levels in a network design also applies to the design of operating systems, database management systems, application software, and distributed systems.

The first initiatives to develop distributed systems, in the 1990s, adopted a limited set of technologies (networking protocols, operating systems, and programming languages), with the objective of minimizing implementation difficulties. This was the case of implementations based on remote procedure calls (RPCs). However, it was still necessary to allow systems that were based on different programming languages or developed under different operating systems to exchange data directly.

The next steps towards ensuring interoperability between heterogeneous systems involved developments ranging from data transfer with format conversion, to the definition of standard interchange formats for specific domains, including geospatial. The World-Wide Web Consortium (W3C), an organization that coordinates the development of popular standards such as the Standard Generalized Markup Language (SGML) and the Hypertext Markup Language (HTML), in 1986 and 1993 respectively, proposed in 1998 the XML as a standard that would serve as the leading exchange format among internet-based applications.

In the field of geographic information systems (GIS), the Open Geospatial Consortium (OGC) was created in 1994, and has been leading some interoperability initiatives in the geospatial domain since then. Among OGC's initiatives is Geography Markup Language (GML), an XML grammar for the specification and encoding of geographic data, published in 2000.

In another line of development, some XML-based protocols were developed with the objective of representing remote method interfaces (WSDL, in 2002), of representing invocations (SOAP, in 2003). Such protocols became the technological base of the first web services.

Scientific Fundamentals

Component-based software development has been the subject of much interest because of its potential to reduce development costs and time, and because of the interest in the deployment of distributed systems. One of the most interesting developments in this field is the emergence of *service-oriented architectures* (SOAs). A SOA is an architecture in which applications are supported through the composition of services, which are well defined, repeatable, tasks that are usually based on data. SOA changes the usual information systems' focus on data towards higher-level processes, in a loosely coupled environment.

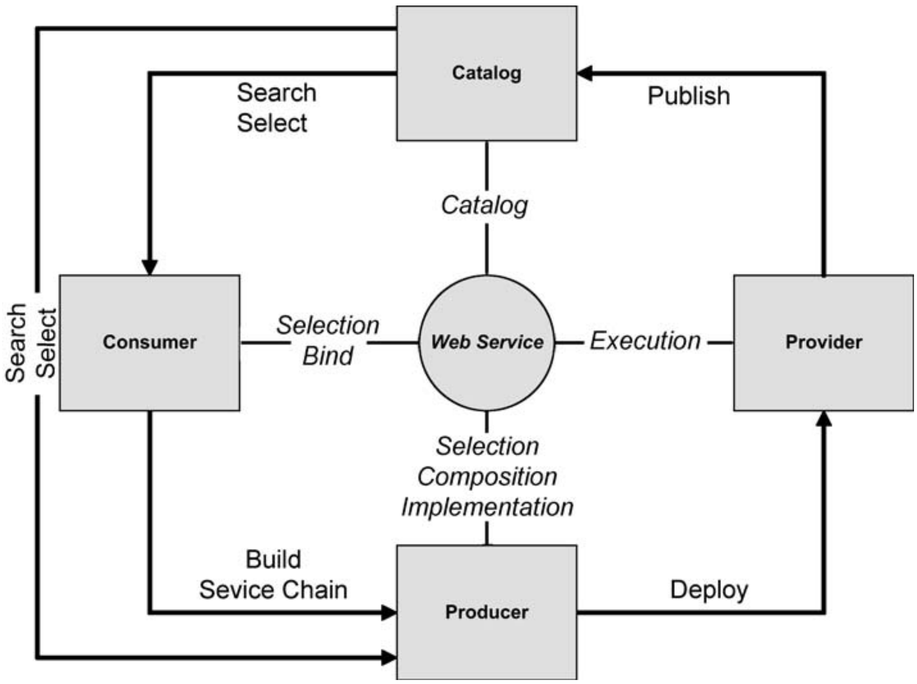
Services, along with their descriptions and fundamental operations, such as *publication, discovery*, *selection* and *binding*, constitute the basis of SOA. SOA supports large applications with sharing of data and processing capacity, through network-based distributed allocation of applications and use of computational resources. In this architecture, services are self-contained, which means that information on the service's description, including its capabilities, interfaces, and behavior, can be obtained from standardized functions contained in the service.

Service *producers*, service *providers*, service *catalogs* and service *consumers* are the actors, or *business entities*, that participate in the conceptual model of web services (Fig. 1). *Publication* is performed by a service provider, and consists in the creation of a service description in WSDL (including several *methods*, comprising the service's functionality) and its publishing on discovery channels on the web. These channels use a registry service to catalog the services, along with some details that allow users to *discover* them. Consumers and producers can then *select* services using a standardized registry service to search through the services catalog and to get the WSDL service descriptor. Using the information in the WSDL descriptor, a client is created for the *binding* operation. At this point, the client is able to initiate a direct communication with the service provider through the Internet, using HTTP and SOAP, and invoking one of the service methods that were described in WSDL through its interface. Binding ends with the reception of the expected web service response in XML or GML.

Service consumers may create software clients which need to access the services through the communication network, and can become producers when they aggregate services from one or more providers into a chain, in order to offer the resulting composite service to their stakeholders. Such chaining of services is transparent to the consumer.

Observe that web services can be implemented, as described, regardless of the programming language options and of the particular computing environment of the services provider. Thus, web services become an interesting alternative for achieving interoperability among information systems.

Using web services in geospatial applications, an interoperability mechanism can be implemented, using geographic data encoding standards (such as GML). This allows for the integration of diverse geographic information sources without the need for the export–transform–import routines that were common in the early days of GIS.



Web Services, Geospatial, Figure 1 Conceptual model of geospatial web services

W

Key Applications

Geospatial web services are useful whenever there is the interest to allow applications to share geographic data available elsewhere. This does not mean that developing geographic or conventional information systems becomes harder, more expensive, or even more complex. The following sample of application areas for geospatial web services is intended to demonstrate that.

Development of Interoperable Systems

In order to ensure that systems are interoperable, it is not necessary to have a system access the other's stored data. On the contrary, it means that functions to access such data, or to use available data to provide required information, can be shared without the need for rewriting code. Thus, the development of new applications becomes simpler and more productive, since (1) no data replication is required, and (2) many functions that would require elaborate coding are already implemented as services. An example is the OGC's Web Coordinate Transformation Service.

Data Sharing

The foremost motivation for the development of geospatial web services is to allow for a direct data exchange between different applications, without requiring a particular understanding of the data source's structure or other implementation details. With geospatial web services, a standardized interface for data access is provided, eliminating the need for the programmer of a client application to know extensive details on how data are encoded and organized in another system.

Code Reuse

Since geospatial web services encapsulate details on how to gain access to data and methods they provide, such information can be fully reused in the development of new information systems. Useful functions provided as methods in a geospatial web service do not need to be reimplemented in the new systems, thus saving on development time.

Spatial Data Infrastructures

Spatial data infrastructures (SDIs) have been defined as the set of technologies, people and policies that are required to facilitate sharing and distribution of geographic information to various levels of government and sectors of society. SDIs can benefit from the uniform and technologically neutral interface provided by geospatial web services, since it is assumed that infrastructure clients are technologically heterogeneous and would be able to choose among several different service providers, thus seeking the best possible arrangement to fulfill their information needs. Notice that, despite the "data" part of the acronym, not only raw data can be provided in an SDI, but also a set of specialized functions that are part of the services. Geospatial web services allow for chaining among these services, so that more specialized responses can be obtained. Geospatial web services also have the advantage of allowing SDI users to keep themselves up-to-date, by providing direct access to the data source, with no need for dataset transfer or temporary data storage.

Privacy Control

Interinstitutional systems provide interfaces based on geospatial web services, which are only allowed to execute methods previously enabled and tested by the service provider. Thus, it is possible to preserve sensitive or confidential data from inadequate use, through controls that reside entirely within the provider organization.

Geospatial Features in General Applications

Conventional information systems can also benefit from the access to geospatial data. Since geospatial functions can be provided in the form of services, conventional systems can have access to geographic functions over geographic data without the need to incorporate GIS tools, or to create a spatial database.

Mobile Geographic Information Clients

Since geospatial web services are remotely processed, clients do not need to store large volumes of data or to perform any complex processing on geographic data. Parameters that characterize the client's geographic information needs are sent to geospatial web services, and a final response, which needs no further processing, is returned to the client. With the current need for urban geographic applications (such as location-based services, mobile computing and others), and the increasing availability of small, low-energy, and internet-ready mobile devices, geospatial web services are becoming an important alternative for a growing information technology market niche.

Future Directions

Virtually any sector of the economy or of the society can benefit from access to correct, up-to-date, and adequate geographic information. Even though there are, traditionally, several sources of basic geographic information, access and integration issues hold many users back, or force more resourceful users to implement complex and problematic data transfer, transformation, and integration routines. Making the access to such information easier, faster, and more practical would contribute to social development and organization, with possible impacts in areas such as transportation, logistics, environmental control, urban planning, and many other business-related activities. In many of those areas, geographic information is an important resource, but companies and individuals are hindered by the fact that they are not able to generate or to obtain basic geographic datasets on their own.

Since sources of geographic data are as varied and heterogeneous as the data itself, geospatial web services provide a means to ensure interoperability among different systems, GIS or conventional information systems, from various organizations.

Therefore, one of the most immediate challenges is to allow for simpler development of new services. This requires, at first, making OGC web service standards more compatible with W3C web services. Complex details on how to obtain geographic data and services should be hidden from the user, and the development of geographic functions in the form of services should be increasingly transparent. In order to accomplish that, it is essential that generic web services and geographic web services can work seamlessly.

Besides, data and interface standards should be open, which would enable the retrieval of geographic information regardless of architectures and computing environment, thus allowing for simpler technological evolution.

Another important issue is the need for increasingly large volumes of quality geographic data. The possibility of having several different and competing data sources to fulfill any specific need is interesting from the standpoint of the user; however, instruments to verify the quality of the information, and to compare it with the user's needs, are still scarce.

Geospatial web services allow us to shift our focus from technological issues to the sociotechnical aspects of the use of geographic information. The expansion of the geospatial data users community will require increasing attention towards a services management model, as well as towards a contents quality model. This direction is compatible with current research and development in the field of Spatial Data Infrastructures, for which geospatial web services are a natural technological choice.

Cross References

► Data Infrastructure, Spatial
► Geography Markup Language (GML)
► Information Services, Geography
► OGC's Open Standards for Geospatial Interoperability
► Web Feature Service (WFS)
► Web Feature Service (WFS) and Web Map Service (WMS)

Recommended Reading

1. OGC website: http://www.opengeospatial.org/
2. Web Services Activity, W3C: http://www.w3.org/2002/ws/
3. OASIS SOA Reference Model Technical Committee: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm
4. International Journal of Spatial Data Infrastructures Research. ISSN 1725-0463. http://ijsdir.jrc.it/
5. Shengru, T., Abdelguerfi, M.: Web services for geographic information systems., IEEE Internet Comput. **10**, 13–15 (2006)

Web Technologies

► Web Mapping and Web Cartography

WebGIS

► University of Minnesota (UMN) Map Server

WFS

► Web Feature Service (WFS)
► Web Feature Service (WFS) and Web Map Service (WMS)

WiMAX

► Indoor Localization
► Intergraph: Real Time Operational Geospatial Applications

Window Query

► R*-tree

WLAN Geolocation

► Indoor Positioning with WirelessLocal Area Networks (WLAN)

WLAN Localization

► Indoor Positioning with WirelessLocal Area Networks (WLAN)

WLAN Location Determination

► Indoor Positioning with WirelessLocal Area Networks (WLAN)

WLAN Location Discovery

► Indoor Positioning with WirelessLocal Area Networks (WLAN)

WLAN Location Estimation

► Indoor Positioning with WirelessLocal Area Networks (WLAN)

WLAN Location Identification

► Indoor Positioning with WirelessLocal Area Networks (WLAN)

WLAN Location Sensing

► Indoor Positioning with WirelessLocal Area Networks (WLAN)

WMS

► deegree Free Software
► OGC's Open Standards for Geospatial Interoperability
► Web Feature Service (WFS)
► Web Feature Service (WFS) and Web Map Service (WMS)
► Web Mapping and Web Cartography

Worlds, Multiple

► Smallworld Software Suite

Wrapper-Mediator Approach to Query Evaluation

► Spatial Information Mediation

XML

► deegree Free Software
► Extensible Markup Language

XML Based Vector Graphics

► Scalable Vector Graphics (SVG)

XML Triple

► Knowledge Representation, Spatial

Z/I Imaging

► Intergraph: Real Time Operational Geospatial Applications

Zernike

► Biomedical Data Mining, Spatial

Zernike Polynomials

► Biomedical Data Mining, Spatial

Bibliography

Aanestad, M., Monteiro, E., Nielsen, P.: Information infrastructures and public goods: Analytical and practical implications for SDI. Inf. Technol. Dev. **13**, 7–25 (2007)

Abdel-Aziz, Y.I., Karara, H.M.: Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. Proc Symp. Close-range Photogrammetry, pp. 1–18, Am. Soc. Photogrammetry (1971)

Abdelmoty, A.I., Smart, P.D., Jones, C.B., Fu, G., Finch, D.: A critical evaluation of ontology languages for geographical information retrieval. J. Visual Languages Comput. **16**, 331–358 (2005)

Abel, D.J., Taylor, K., Ackland, R., Hungerford, S.: An exploration of gis architectures for internet environments. Comp Env Urban Sys **22**(1), 7–23 (1998)

Abhinandan, D., Gehrke, J., et al.: Approximation techniques for spatial data. Proceedings of the 2004 ACM SIGMOD international conference on Management of data. Paris, France. SESSION: Research sessions: spatial data. Pgs. 695–706. (2004)

Abiteboul, S., Hull, R.: IFO: A formal semantic database model. ACM Trans. Database Syst. **12**, 525–565 (1987)

Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley, Reading, Massachusetts (1995)

Abowd, G.D., Atkeson, C.G., Hong, J., Long, S., Kooper, R., Pinkerton, M.: Cyberguide: A mobile context-aware tour guide. Wireless Networks **3**(5), 421–433 (1997)

Abowd, G.D., Mynatt, E.D., Rodden, T.: The human experience. IEEE Pervasive Comput. **1**, 48–57 (2003)

Abraham, T., Roddick, J.F.: Survey of spatio-temporal databases. Geoinformatica 3(1): 61–99 (1999)

Acharya, S., Gibbons, P., Poosala, V., Ramaswamy, S.: Joint synopses for approximate query answering. In: SIGMOD '99: Proceedings of the 1999 ACM SIGMOD international conference on Management of data, pp. 275–286. ACM Press, New York, NY, USA (1999)

Ackoff, R.L.: From Data to Wisdom, J. Appl. Syst. Anal. **16**, 3–9 (1989)

Adam, N., Janeja, V.P., Atluri, V.: Neighborhood based detection of anomalies in high dimensional spatio-temporal sensor datasets. In: 19th ACM Symp. on Applied Computing (2004)

ADHS Guides to Good Practice http://ahds.ac.uk/history/creating/guides/gis/ Accessed 21 March 2007

Aerts, J.C.J.H., Goodchild, M.F., Heuvelink, G.B.M.: Accounting for spatial uncertainty in optimization with spatial decision support systems. Trans. GIS **7**(2), 211–230 (2003)

Afrati, F., Cosmadakis, S., Grumbach, S., Kuper, G.: Linear versus polynomial constraints in database query languages. In: Borning, A. (ed.) Proceedings of the 2nd Workshop on Principles and Practice of Constraint Programming. Lecture Notes in Computer Science, vol. 874, pp. 181–192 Springer-Verlag, Heildelberg (1994)

Agarwal, P.: Contested Nature of place. In: Egenhofer, M., Freska, C., Harvey, M. (eds.) 3rd International Conference GIScience, 2004, LNCS 3234, pp. 1–21. Springer, Berlin (2004)

Agarwal, P., Har-Peled, S.: Maintaining Approximate Exten Measures of Moving Points. In: Proceedings of the 12th ACM-SIAM Sympos. Discrete Algorithms, Washington, DC, USA, 7–9 Jan 2001, pp. 148–157

Agarwal, P., Huang, Y., Dimitrova, V.:, Formal approach to reconciliation of individual ontologies for personalization of geospatial semantic web. In: Proceedings of the First International Conference on Geospatial Semantics, Mexico City, November 2005. Lecture Notes in Computer Science, vol. 3799, pp.202–219, Springer-Verlag, Berlin (2005)

Agarwal, P.K., Arge, L., Erickson, J.: Indexing moving points. J. Comput. Syst. Sci. **66**(1), 207–243 (2003)

Agarwal, P.K., Arge, L., Erickson, J.: Indexing Moving Points. In: Proceedings of the 19th ACM Symp. on Principles of Database Systems, Dallas, Texas, USA, 15–17 May 2000, pp. 175–186

Agarwal, P.: Ontological considerations in GIScience. Int. J. Geogr. Inf. Sci. **19**, 501–535 (2005)

Ageenko, E.I., Frnti, P.: Compression of large binary images in digital spatial libraries. Comput. Graphics **24**, 91–98 (2000)

Aggarwal, C.C., Agrawal, D.: On nearest neighbor indexing of nonlinear trajectories. In: PODS, pp. 252–259 (2003)

Aggarwal, C., Hinneburg, A., Keim, D.A.: On the surprising behavior of distance metrics in high dimensional space. In: Proceedings of the 8th International Conference on Database Theory, London, 6 January 2001, pp. 420–434

Aggett, G., McColl, C.: Breaking down barriers: geospatial decision-support systems help citizens embrace community land-use planning. Land Bulletin of the National Consortium for Rural Geospatial Innovations. University of Wisconson, Madison, WI (2006) www.ruralgis.org

Aggett, G.R., Wilson, J.P.: Creating and coupling a high-resolution DEM with a 1-D hydraulic model in a GIS for scenario-based assessment of avulsion in a gravel-bed river. Geomorphology, in review (2007)

Agouris, P., Stefanidis, A., Gyftakis, S.: Differential Snakes for Change Detection in Road Segments. Photogramm. Eng. Remote Sens. **67**(12), 1391–1399 (2001)

Agrawaland, R., Srikant, R.: Mining sequential patterns. In: Proc. of SIGKDD (1995)

Agrawal, R., Faloutsos, C., Swami, A.: Efficient Similarity Search In Sequence Databases. In: Lecture Notes in Computer Science, vol. 730, Berlin/Heidelberg (1993)

Agrawal, R., Skrikant, R.: Fast algorithms for mining association rules. In: Proceedings of the 20th Intnational Conference on Very Large Data Bases, pp. 487–499 (1994)

Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: Proc. of the ACM SIGMOD Conference on Management of Data, May 2000, pp. 439–450. ACM Press, Dallas, Texas (2000)

Aguirre, A.A., Ostfeld, R.S., Tabor, G.M., House, C., Pearl, M.C.: Conservation Medicine: Ecological Health in Practice. Oxford University Press, Oxford (2002)

Ahern, S., Davis, M., King, S., Naaman, M., Nair, R.: Reliable, User-Contributed GSM Cell-Tower Positioning Using Context-Aware Photos. In: The Eighth International Conference on Ubiquitous Computing (UbiComp) (2006)

Ahlqvist, O.: Using uncertain conceptual spaces to translate between land cover categories. Int. J. Geogr. Inf. Sci. **19**, 831–85 (2005)

Ahl, V., Allen, T.F.H.: Hierarchy Theory – A Vision, Vocabulary, and Epistemology. Columbia University Press, New York (1996)

Ahonen-Rainio, P., Kraak M.J.: Deciding on Fitness for Use: Evaluating the Utility of Sample Maps as an element of Geospatial Metadata. Cartography and Geographic Information Science **32**(2), 101–112 (2005)

Air Release (AIRS/AFS). http://www.epa.gov/enviro/html/airs/index.html. Accesed 20 Aug 2007

Akenine-Möller, T., Haines, E.: Real-Time Rendering, A K Peters Ltd, Wellesley, MA (2002)

Akinlar, C., Aref, W.G., Kamel, I., Mukherjee, S.: Automatic disks: The building block for a scalable distributed file system. In: Proc. of the 5th Intl. Workshop on Multimedia Information Systems, MIS, Palm Springs Desert, October 1999

Alavi, B., Pahlavan, K.: Indoor geolocation distance error modeling with UWB channel measurements. In: Proceedings of the IEEE Personal Indoor Mobile Radio Communications Conference (PIMRC), Berlin, 11–14 September 2005

Albert, R., Barabasi, A.: Statistical mechanics of complex networks. Rev. Mod. Phys. **74**, 47–97 (2002)

Albertz, J.: 90 Jahre Deutsche Gesellschaft für Photogrammetrie und Fernerkundung e.V. Photogrammetrie-Fernerkundung-Geoinformation PFG 5(1999), pp. 293–349 (1985)

Albowicz, J., Chen, A., Zhang, L.: Recursive position estimation in sensor networks. In: IEEE Int. Conf. Net. Proto. Nov 2001, pp. 35–41 (2001)

Alexander, C., Ishikawa, S., Silverstein, M.: A Pattern Language – Towns, Buildings, Construction. Oxford University Press, New York (1977)

The Algorithm Design Manual. Nearest Neighbor Search. Jun. 2 (1997)

Ali, M., Goovaerts, P., Nazia, N., Haq, M.Z., Yunus, M., Emch, M.: Application of Poisson kriging to the mapping of cholera and dysentery incidence in an endemic area of Bangladesh. Int. J. Health Geogr., **5**, article 45 (2006)

Ali, W., Moulin, B.: 2D-3D Multi-Agent Geo-Simulation with Knowledge-Based Agents of Customers' Shopping Behavior in a Shopping Mall. In: Cohn, A.G., Mark, D.M. (eds.) Spatial Information Theory, p. 445–458. Springer, Berlin (2005)

Allen, G.: Applied Spatial Cognition: From Research to Cognitive Technology. Lawrence Erlbaum Associates, Mahwah NJ (2007)

Allen, G.L.: Spatial abilities, cognitive maps, and wayfinding: Bases for individual differences in spatial cognition and behavior. In: Golledge R.G. (ed.) Wayfinding behavior: Cognitive mapping and other spatial processes, pp. 46–80. The Johns Hopkins University Press, Baltimore (1999)

Allen, J.F.: Maintaining knowledge about temporal intervals. In: Communications of the ACM, **26**(11), pp. 832–843. ACM Press, New York, NY, USA (1983)

Allen, J.: Towards a general theory of action and time. Artif. Intell. **23**, 123–54 (1984)

Almeida, V., Güüting, R.: Supporting uncertainty in moving objects in network databases. In: Proc. GIS, 2005

Almeida, V.T., Güting, R.H., Behr, T.: Querying moving objects in SECONDO. In: Proceedings of the Mobile Data Management Conference, Nara, Japan, pp. 47–51 (2006)

de Almeida, V.T., Güting, R.H.: Indexing the trajectories of moving objects in networks. GeoInformatica, **9**(1) (2005).

De Almeida, V.T., Güting, R.H.: Using Dijkstra's Algorithm to Incrementally Find the k-Nearest Neighbors in Spatial Network Databases, In: Proc. ACM Symposyum on Apagesied Computing, pp. 58–62 (2006)

Alsnih, R., Stopher, P.R.: Review of Procedures Associated With Devising Emergency Evacuation Plans. Transportation Research Record. pp. 89–97. Transportation Research Board, Washington, DC (2004)

Altheide, P.: GIS Data Conversions. Strategies, Techniques, Management, (ed.) Pat Hohl. Chapter 15 Spatial Data Transfer Standards, pp. 318–354, Onword Press, Santa Fe, NM (1998)

Altschul, S. F., Gish, W., Miller, W., Myers, E.W., Lipman D.J.: A basic local alignment search tool. Journal of Molecular Biology **215**(3), 403–410 (1990)

Amaral, S., Monteiro, A.M.V., Câmara, G., Quintanilha, J.A.: Evolution of the Urbanization Process in the Brazilian Amazonia In: Levine, N. (ed.) CrimeStat III: A Spatial Statistics Program for the Analysis of Crime Incident Locations (version 3.0), Chapter 8. Ned Levine & Associates and the National Institute of Justice, Houston, TX, Washington, DC (2004)

American Congress on Surveying and Mapping: The Proposed Standard for Digital Cartographic Data. Am. Cartogr. **15**(1), entire issue (1988)

American Forests: CITYgreen: Calculating the Value of Nature, Version 5.0 User's Manual, American Forests, Washington, D.C., See also http://www.americanforests.org/productsandpubs/citygreen/ (2002)

American National Standards Institute, National Committee for Information Technology Standards: ANSI NCITS 320-1998, Spatial Data Transfer Standard Parts 1–4

Ames, D.P.: Bayesian Decision Networks for Watershed Management. Utah State University, Logan, UT (2002)

Ames, D.P., Neilson, B.T., Stevens, D.K., Lall, U.: Using Bayesian networks to model watershed management decisions: an East Canyon Creek case study. IWA Publishing, J. Hydroinform. (2005)

Ameskamp, M.: Three-dimensional Rule-based Continuous Soil Modelling. PhD thesis, Christian-Albrechts-Universitaet Kiel (1997)

Analytical CRM software opens up new worlds to marketers, by Paul Thompson, June 17 (2002) http://www.strategymag.com/articles/d+i/20020617/dicrmanalytical.html?word=Analytical&word=CRM&word=Software

Anders, K.-H., Bobrich, J.: MRDB Approach for Automatic Incremental Update. Paper presented at the ICA Workshop on Generalization and Multiple Representation, Leicester, 20–21 August 2004. Available via: http://ica.ign.fr/Leicester/paper/Anders-v1-ICAWorkshop.pdf. Accessed 12 Feb 2007

Anderson, J.R., Hardy, E.E., Roach, J.T., Witmer, R.E.: A Land Use and Land Cover Classification System For Use With Remote Sensor Data. USGS Professional Paper **964** (1976)

Anderson, N.H., Titterington, D.M.: Some methods for investigating spatial clustering, with epidemiological applications. J. R. Stat. Soc. **160**, 87–105 (1997)

Anderson, S.: Aggregation estimation for 2D moving points. 13th International Symposium on Temporal Representation and Reasoning, IEEE Press, Washington (2006)

Andersson, M., Gudmundsson, J., Laube, P., Wolle T.: Reporting leadership patterns among trajectories. In: Proceedings of the 2007 ACM Symposium on Applied Computing, pp. 3–7. ACM Press, New York (2007)

Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In: Proceedings of the Symposium on Foundations of Computer Science, pp. 459–468 (2006)

Andrienko, N.V., Andrienko, G.L.: Interactive maps for visual data exploration. Int. J. Geogr. Inf. Sci. **13**(4), 355–374 (2003)

Anez, J., de la Barra, T., Perez, B.: Dual Graph Representation of Transport Networks. Transp. Res. **30**(3), 209–216 (1996)

Ang, C.-H., Tan, T.C.: New linear node splitting algorithm for r-trees. In: Proc. of Symposium on Advances in Spatial Databases (SSD), Berlin, Germany, 15–18 July 1997, pp. 339–349

Anne, M., Crowley, J.L., Devin, V., Privat, G.: Localisation intra-bâtiment multitechnologies: RFID, Wi-Fi, vision. In: Proceedings of Ubimob 2005, Grenoble, France, 2005, Vol.120, pp. 29—35

Anonymous: The free software definition. http://www.fsf.org/licensing/essays/free-sw.html (2006). Accessed 26 Apr 2006

Anonymous: Sequence alignment software. http://en.wikipedia.org/wiki/Sequence_alignment_software (2007). Accessed

Anonymous: VT Conference Puts New Research Area on the Map; GIS Expert Michael Goodchild Echoes Its Value. https://www.vbi.vt.edu/article/articleview/48/1/15/ (2007). Accessed 05 March 2007

Anselin, L.: Exploratory spatial data analysis and geographic information systems. In: New Tools for Spatial Analysis, p. 45–54. Eurostat, Luxemburg (1994)

Anselin, L.: Local indicators of spatial association: Lisa. Geogr. Anal. **27**(2):93–115 (1995)

Anselin, L.: Spatial Econometrics: Methods and Models. Kluwer Academic, Dordrecht (1988)

Anselin, L.: Under the hood Issues in the specification and interpretation of spatial regression models. Agricul. Econ. **27**(3), 247 (2002)

Anselin, L.: Under the Hood: Issues in the Specification and Interpretation of Spatial Regression Models. http://sal.uiuc.edu/users/anselin/papers.html (2002)

Antoniou, G., Harmelen, F.: A Semantic Web Primer. MIT Press, Cambridge, MA (2004)

Aoki., P.M.: Generalizing "search" in generalized search trees (extended abstract). In: ICDE, pp. 380–389 (1998)

Apple Developer Connection: What Is Performance? (2006) http://developer.apple.com/documentation/Performance/Conceptual/PerformanceOverview/DevelopingForPerf/chapter_2_section_2.html. Accessed 29 Aug 2007

Apple. Human Interface Guidelines: The Apple Desktop Interface, Apple Computer, Inc. Addison-Wesley Publishing, Reading, MA (1987)

Appletgate, D., Bixby, R., Chvátal, V., Cook, W.: On the solution of traveling salesman problems. Documenta Mathematica Journal der Deutschen Mathematiker-Vereinigung International Congress of Mathematicians, (1998)

Arampatzis, A., van Kreveld, M., Reinbacher, I., Jones, C.B., Vaid, S., Clough, P.D., Joho, H., Sanderson, M.: Web-based delineation of imprecise regions. J. Comput. Environ. Urban Syst. (CEUS) **30**(4) (2006)

Arasu, A., Babcock, B., Babu, S., Datar, M., Ito, K., Nishizawa, I., Rosenstein, J., Widom, J.: STREAM: The Stanford stream data manager. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, California, 2003, p. 665

Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A., Raghavan, S.: Searching the web. ACM Trans. Internet Technol. **1**, 2–43 (2001)

Arbia, G., Griffith, D.A., Haining, R.P.: Spatial error propagation when computing linear combinations of spectral bands: the case of vegetation indices. Environ. Ecol. Stat. **10**, 375–396 (2003)

Arbia, G.: Statistical effect of data transformations: a proposed general framework. In: Goodchild, M.F., Gopal, S. (eds.) The accuracy of spatial databases, 249–259. Taylor & Francis, London (1989)

Ardagna, C.A., Cremonini, M., Damiani, E., De Capitani di Vimercati, S., Samarati, P.: Supporting location-based conditions in access control policies. In: Proceedings of ASIACCS'06, March 21–24, 2006, Taipei, Taiwan

Ardekani, S.A.: A Decision Tool for Transportation Operations Following Urban Disasters, Final Report. National Science Foundation. The University of Texas at Arlington, Arlington (1992)

Ardekani, S.A., Hobeika, A.: Logistics Problems in the Aftermath of the 1985 Mexico City Earthquake. Transp. Q. **42**, 107–124 (1988)

Aref, W.G., Ilyas, I.F.: An extensible index for spatial databases. In: SSDBM, pp. 49–58 (2001)

Aref, W.G., Kamel, I., Ghandeharizadeh, S.: Disk scheduling in video editing systems. IEEE Trans. Knowl. Data Eng., TKDE **13**(6),933–950 (2001)

Argyle, M.: Bodyly Communication. Methuen & Co., London (1975)

Armitage, L.H., Enser, P.G.B.: Analysis of user need in image archives. J. Inf. Sci. **23**(4): 287–299 (1997)

Armstrong, M.P., Cowles, M.K., Wang, S.: Using a computational grid for geographic information analysis: a reconnaissance. Prof. Geogr. **57**, 365–375 (2005)

Armstrong, A.P., Densham, P.J.: Database organization strategies for spatial decision support systems. Int. J. Geogr. Inf. Sys. **4**(1), 3–20 (1990)

Arora, S.: Polynomial-Time Approximation Schemes for Euclidean TSP and other Geometric Problems. J. ACM **45**(5), 753–782 (1998)

Arpinar, A., Sheth, A., Ramakrishnan, C.: Geospatial Ontology Development and Semantic Analytics. In: Wilson, J.P., Fotheringham, A.S. (eds.) Handbook of Geographic Information Science, Blackwell Publishing, Boston (2004)

Arthur, P., Passini, R.: Wayfinding: People, Signs, and Architecture, 238. McGraw-Hill Ryerson, Toronto (1992)

Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans. Signal Process. **50**, 174–188 (2002)

Arunasalam, B., Chawla, S., Sun, P.: Striking two birds with one stone: Simultaneous mining of positive and negative spatial patterns. In: Proceedings of 2005 SIAM International Conference on Data Mining SDM05, to appear (2005)

Arya, S., Mount, D., Netanyahu, N., Silverman, R., Wu, A.: An optimal algorithm for approximate nearest neighbor searching. J. ACM **45**(6), 891–923 (1998)

Asano, T., Ranjan, D., Roos, T., Welzl, E., Widmayer, P.: Space-filling curves and their use in the design of geometric data structures. Theor. Comput. Sci., TCS **181**(1), 3–15 (1997)

Asher, N., Lang, J.: When nonmonotonicity comes from distance. In: Nebel, B., Dreschler-Fischer, L. (eds.): KI-94: Advances in

Artificial Intelligence. pp. 308–318. Springer-Verlag, Heidelberg (1994)

Ash, J.N., Moses, R.L.: Outlier compensation in sensor network self-localization via the EM algorithm. In: IEEE ICASSP, March 19–23, (2005)

Ash, J.N., Potter, L.C.: Sensor network localization via received signal strength measurements with directional antennas. In: Proceedings of the 2004 Allerton Conference on Communication, Control, and Computing (2004)

Ashraful, A., Thuraisingham, B.: Geospatial resource description framework (GRDF) and secure GRDF. Tech. Rep. UTDCS-03–06, University of Texas at Dallas (2006) http://www.cs.utdallas.edu

Asif, Z.: Integrating the supply chain with RFID: A technical and business analysis. Commun. Assoc. Inf. Syst. **15**, 393–427 (2005)

Asrar, G., Dozier, J.: EOS: Science Strategy for the Earth Observing System.American Institute of Physics Press, Woodbury, New York (1994)

Asrar, G., Greenstone, R.: MTPE EOS Reference Handbook, NASA Goddard Space Flight Center, Greenbelt, MD (1995)

Assessing the Homeland Security Implications of Publicly Available Geospatial Information. Rand Report for NGA (2004)

Asthana, A., Cravatts, M., Krzyzanowski, P.: An indoor wireless system for personalized shopping assistance. In: IEEE Workshop on Mobile Computing Systems and Applications, pp. 69–74. Santa Cruz, California (1994)

Athitos, V.: Approximate Nearest Neighbor Retrievel Using Euclidean Embeddings. Approximate Nearest Neighbors Methods for Learning and Vision. Dec 13 (2003)

Atkinson, K.B. (ed.): Close Range Photogrammetry And Machine Vision. Whittles, Caithness, UK (1996)

Atkinson, P.M., Webster, R., Curran, P.J.: Cokriging with airborne MSS imagery. Remote Sens. Environ. **50**, 335–345 (1994)

Atkinson R, Fitzke J (ed) (2006) Gazetteer Service Profile of the Web Feature Service Implementation Specification. OpenGIS Project Document 05-035r2. http://portal.opengeospatial.org/files/?artifact_id=15529

Atluri, V., Adam, N., Youssef, M.: Towards a unified index scheme for mobile data and customer profiles in a location-based service environment. In: Proceedings of theWorkshop on Next Generation Geospatial Information (NG2I'03), Cambridge, MA, USA 19–21 Oct 2003

Atluri, V., Chun, S.: An authorization model for geospatial data. IEEE Trans. Dependable Sec. Comput. **1**, 238–254 (2004)

Atluri, V., Chun, S.A.: A geotemporal role-based authorization system. Int. J. Inf. Comput. Security (in press, 2007)

Atluri, V., Guo, Q.: STAR-Tree: An index structure for efficient evaluation of spatiotemporal authorizations. In: IFIP TC11/WG 11.3 Eighteenth Annual Conference on Data and Applications Security, Sitges, Catalonia, Spain, 25–28 2004

Atluri, V., Guo, Q.: Unified index for mobile object data and authorizations. In: ESORICS, Milan, Italy, 12–14 2005

Atluri, V., Mazzoleni, P.: Uniform indexing for geospatial data and authorizations. In: DBSec, Kings College, Cambridge, UK, 28–31 July 2002

Atluri, V., Shin, H.: Efficient enforcement of security policies based on tracking of mobile users. In: DBSec, Sophia, Antipolis, France, 31 Aug 2006

Augustinraj, R.: A Study of Spatiotemporal Health Effects due to Water Lead Contamination. M.S. Thesis, University of North Carolina (2002)

Aurenhammer, F.: Voronoi diagrams survey of a fundamental geometric data structure. In: ACM Computing Surveys, vol. 23, pp. 345–405. (1991)

Autodesk Geospatial (http://images.autodesk.com/adsk/files/autodesk_geospatial_white_paper.pdf)

Autodesk, Inc.: Map 3D 2007 Essentials. Autodesk Press, San Rafael (2007)

Avery, T.E., Berlin, G.L.: Interpretation of Aerial Photographs. 4th edn. Burgess Publishing, Minneapolis, MN (1985)

Avoine, G., Oechslin, P.: Financial Cryptography and Data Security, In: RFID traceability: A multilayer problem, Lecture Notes in Computer Science, vol. 3570, pp. 125–140. Springer (2005)

Axelrod, R.: The Evolution of Cooperation. Basic Books, Princeton (1997)

Aydin, G., Sayar, A., Gadgil, H., Aktas, M.S., Fox, G.C., Ko, S., Bulut, H., Pierce, M.E.: Building and applying geographical information system grids. In: Proceedings of GGF15 Workshop, Boston, MA, 3–6 Oct 2005

Babcock, G., Babu, S., Datar, M., Motwani, R., Widom, J.: Models and issues in data stream systems. In: ACM Symposium on Principles of Database Systems (PODS), Madison, Wisconsin, 2002, pp. 1–16

Badard, T., Braun, A.: OXYGENE: A platform for the development of interoperable geographic applications and web services. In: Proceedings of the 15th International Workshop on Database and Expert Systems Applications (DEXA'04), Zaragoza, Spain, 30 Aug–3 Sept 2004

Baddeley, A., Turner, R.: Spatstat: An r package for anayzing spatial point patterns. J. Stat. Softw. **12**, 1–42 (2003)

Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co. Inc., Boston, MA, USA (1999)

Bahl, P., Padmanabhan, V.N.: RADAR: An in-building RF-based user location and tracking system. In: Proceedings of the IEEE INFOCOM 2000, Tel Aviv, Israel, 26–30 March 2000

Bailey-Kellogg, C., Ramakrishnan, N., Marathe, M.V.: Spatial Data Mining for Pandemic Preparedness. ACM SIGKDD Explorations **8**(1), 80–82 (2006)

Bailey, T.C., Gatrell, A.C.: Interactive Spatial Data Analysis. Prentice Hall, Harlow, UK (1995)

Bakeman, R., Gottman, J.M.: Observing Interaction. Cambridge University Press, Cambridge, New York (1997)

Baker, J.C., Lachman, B.E., Frelinger, D.R., O'Connell, K., Hou, A., Tseng, M., Orletzky, D., Yost, C.: Mapping the Risks: Assessing the Homeland Security Implications of Publicly Available Geospatial Information. Rand Corporation, Santa Monica, CA (2004)

Baker, J.C., O'Connell, K.M., Willamson, R.A. eds:. Commercial Observation Satellites - At the Leading Edge of Global Transparency, Rand Corporation:Arlington, VA (2001)

Baker, W.L., Weisberg, P.J.: Using GIS to model tree population parameters in the rocky mountain national park forest-tundra ecotone. J. Biogeogr. **24**, 513–526 (1997)

Balakrishnan, P., et al.: Comparative performance of the FSCL neural net and K-means algorithm for market segmentation. Eur. J. Oper. Res. **93**(2), 346–357 (1996)

Bales, R.F.: Die Interaktionsanalyse. Ein Beobachtungsverfahren zur Untersuchung kleiner Gruppen. In: König, R. (ed.) Praktische Sozialforschung II. Beobachtung und Experiment in der Sozialforschung, pp. 148–167. Kiepenheuer & Witsch, Köln, Berlin (1962)

Bales, R.F.: Interaction process analysis. Cambridge/Mass., Cambridge, New York (1951)

Balley, S., Parent, C., Spaccapietra, S.: Modelling geographic data with multiple representation. International Journal of Geographical Information Science **18**(4), 327–352 (2004)

Balram, S., Dragicevic, S.: Collaborative Geographic Information Systems: Origins, Boundaries, and Structures. In: Balram, S., Dragicevic, S. (eds.) Collaborative Geographic Information Systems, pp. 1–22. Idea Group Publishers, Hershey, PA (2006)

Baltsavias, E.: Object extraction and revision by image analysis using existing geodata and knowledge: current status and steps towards operational systems. ISPRS J. Photogramm. Remote Sensing **58**, 129–151 (2004)

Banerjee, S., Carlin, B.P., Gelfand, A.E.: Hierarchical Modeling and Analysis for Spatial Data. Chapman & Hall/CRC, London, Boca Raton (2004)

Banerjee, S., Finley, A.O.: Bayesian multi-resolution modelling for spatially replicated datasets with application to forest biomass data. J. Statistical Plann. Inference (2007). doi: 10.1016/j.jspi.2006.05.024

Banerjee, S., Johnson, G.A.: Coregionalized single and multiresolution spatially varying growth-curve modelling. Biometrics, **62**, 864–876 (2006)

Banerjee, S.: Revisiting Spherical Trigonometry with Orthoginal Projectors, College Math. J. **35**(5), 375–381 (2004)

Bansal, N., Blum, A., Chawla, S., Meyerson, A.: Approximation Algorithms for Deadline-TSP and Vehicle Routing with Time-Windows. In: Proc. of FOCS (2004)

Baorzsaonyi, S., Kossmann, D., Stocker, K.: The Skyline Operator. ICDE (2001) pp. 421–430, Heidelberg, Germany

Baraldi, A., Blonda, P.: A survey of fuzzy clustering algorithms for pattern recognition. IEEE Transactions on Systems, Man and Cybernetics Parts I and II **29**(6), 778–785 (1999)

Barber, S. Beyond Performance Testing Part 2: A Performance Engineering Strategy (2004) http://www-128.ibm.com/developerworks/rational/library/4215.html. Accessed 29 Aug 2007

Barclay T, Gray J,Slutz D, (1999) Microsoft TerraServer: A Spatial Data Warehouse, Microsoft Technical Report, http://research.microsoft.com/users/Tbarclay/MicrosoftTerraServer_TechnicalReport.doc. Accessed 20 Aug 2006

Bär, H.R., Sieber, R.: Towards high standard interactive atlases: the GIS and multimedia cartography approach. In: Proceedings of the 19th International Cartographic Conference, Ottawa, 14–21 August 1999, pp. 235–241

Barnett, V., Lewis, T.: Outliers in Statistical Data. John Wiley, New York (1994)

Barrault, M., Regnauld, N., Duchene, C., Haire, K., Baeijs, C., Demazeau, Y., Hardy, P., Mackaness, W., Ruas, A., Weibel, R.: Integrating multi agent, object oriented and algorithmic techniques for improved automated map generalization. In: Proceedings of the 20th International Cartographic Conference, **3**, 2110–2116, Beijing, China (2001)

Barrett, C., Beckman, R., Berkbigler, K., Bisset, K., Bush, B., Campbell, K., Eubank, S., Henson, K., Hurford, J., Kubicek, D., Marathe, M., Romero, P., Smith, J., Smith, L., Speckman, P., Stretz, P., Thayer, G., Eeckhout, E., Williams, M.D.: TRANSIMS: Transportation Analysis Simulation System. Technical Report LA-UR-00–1725, Los Alamos National Laboratory Unclassified Report. An earlier version appears as a 7 part technical report series LA-UR-99-1658 and LA-UR-99-2574 to LA-UR-99–2580 (2001)

Barrett, C., Eubank, S., Marathe, M.: Modeling and Simulation of Large Biological, Information and Socio-Technical Systems: An Interaction Based Approach, to appear. In: Goldin, D., Smolka, S., Wegner, P. (eds.) Interactive Computation: The New Paradigm. Springer, New York (2005)

Barrett, C., Smith, J.P., Eubank, S.: Modern Epidemiology Modeling. Sci. Am. **292**(3), 54–61 (2005)

Barry, D.: Web Services and Service-Orientated Architectures, The Savvy Manager's Guide, Your Road Map to Emerging IT. Morgan Kaufmann Publishers, San Francisco, CA. (2003)

Barry, R.P.: The local approximation of variograms in R^2 by covariance functions. Stat. Probab. Lett. **74**, 171–177 (2005)

Bar-Shalom, Y., Fortmann, T.E.: Tracking and data association. Mathematics in Science and Engineering Series 179. Academic, San Diego (1988)

Barvinok, A., Fekete, S., Johnson, D., Tamir, A., Woeginger, G., Woodroofe, R.: The geometric maximum traveling salesman problem. J. ACM (JACM) **50**:(5)3, (2003)

Barzoharand, M., Cooper, D.B.: Automatic Finding of Main Roads in Aerial Images by Using Geometric Stochastic Models and Estimation. IEEE Transaction on Pattern. Anal. Mach. Intell. **18**(7), 707–721 (1996)

Basch, J., Guibas, L.J., Hershberger, J.: Data structures for mobile data. In: Proceedings of ACM-SIAM SODA, pp. 747–756. New Orleans, Lousiana, USA, 5–7 Jan 1997

Batini, C., Lenzerini, M., Navathe, S.B.: A comparative analysis of methodologies for database schema integration. ACM Comput. Surv. **18**, 323–364 (1986)

Battiti, R., Brunato, M., Villani, A.:. Statistical learning for location fingerprinting in wireless LANs. In: *Technical Report DIT-02–0086*, University of Trento, Informatic and Telecommunication Department, Italy (2002)

Batty, M.: Agents, cells, and cities: new representational models for simulating multiscale urban dynamics. Environ. Planning A **37**, 1373–1394 (2005)

Batty, M.,Desyllas, J., Duxbury, E.: The discrete dynamics of small-scale spatial events: agent-based models of mobility in carnivals and street parades. Int. J. Geogr. Inf. Sci. **17**(7), 673–697 (2003)

Batty, M.: Planning support systems: technologies that are driving planning. In: Geertman, S., Stillwell, J. (eds.) Planning Support Systems in Practice, pp. v-viii. Springer, New York, NY (2003)

Batty P. 2004: Future Trends & the Spatial Industry – Part Two. Geospatial Solutions. Advanstar Communications, New York, September, pp. 32-35

Bauer, M., Fabian, B., Fischmann, M., Gürses, S.: Emerging Markets for RFID Traces, Arxiv preprint cs.CY/0606018 (2006)

Bayardo Jr., R.J., Agrawal, R.: Data Privacy through Optimal *k*-Anonymization. In: Proceedings of the International Conference on Data Engineering (ICDE), pp. 217–228. Tokyo, Japan (2005)

Bayer, R., McCreight, E.M.: Organization and maintenance of large ordered indexes, pp. 107–141, SIGFIDET Workshop (1970)

Bayer, R., McCreight, E.: Organization and maintenance of large ordered indexes. In: Software Pioneers - Contributions to software engineering, pp. 245–262 (2002)

Beaujardiere, J. de la (eds.) (2006) OpenGIS® Web Maper Server Implementation Specification, OGC doi:06-042 http://portal.opengeospatial.org/files/?artifact_id=14416

Beaujardiere, J.: Web Mapping Service, OGC Implémentation Spécification http://www.portal.opengoespatial.org/files/?artifact_id=14416(2004). Accesed 9 Aug 2007

Beccaria, C.: in Richard Davies, translator: On Crimes and Punishments, and Other Writings. Cambridge University Press (1764)

Becker, B., Gschwind, S., Ohler, T., Seeger, B., Widmayer, P.: An asymptotically optimal multiversion B-Tree. VLDB J. **5**(4), 264–275 (1996)

Beckmann, N., Kriegel, H.-P., Schneider, R., Seeger, B.: The R*-tree: An efficient and robust access method for points and rectangles. In: Proc. ACM International Conference on Management of Data (SIGMOD'90), pp. 322–331. ACM Press, New York, NY, USA (1990)

Beckman, R., Baggerly, K., McKay, M.: Creating synthetic base-line populations. Transportation Research Part A, Policy Pract. **30**, 415–429 (1996)

Beckman, R.J., et al: TRANSIMS-Release 1.0: The Dallas-Fort Worth Case Study, Technical Report LA-UR-97-4502, Los Alamos National Laboratory (1997)

Bédard, Y., Bernier, É.: Supporting multiple representations with spatial databases view management and the concept of VUEL. In: The Joint Workshop on Multi-Scale Representation of Spatial Data, ISPRS WG IV/3, ICA Commission. On Map Generalization, Ottawa, Canada, 7–8 July 2002

Bédard, Y., Caron, C., Maamar, Z., Moulin, B., Vallière, D.: Adapting Data Model for the Design of Spatio-Temporal Database. Comput., Environ. Urban Syst. **20**(l), 19–41 (1996)

Bédard, Y., Larrivée, S.: Développement des systèmes d'information à référence spatiale: vers l'utilisation d'ateliers de génie logiciel. CISM Journal ACSGC, Canadian Institute of Geomatics, Canada, **46**(4), 423–433 (1992)

Bédard, Y., Larrivée, S., Proulx, M.-J., Nadeau, M.: Modeling Geospatial Databases with Plug-Ins for Visual Languages: A Pragmatic Approach and the Impacts of 16 Years of Research and Experimentations on Perceptory. In: Wang, S. et al. (eds.): Conceptual Modeling for Advanced Application Domains. Lecture Notes in Computer Science, Vol. 3289, pp. 17–30. Springer-Verlag, Berlin/Heidelberg (2004)

Bédard, Y., Merrett, T., Han, J.: Fundamentals of geospatial data warehousing for geographic knowledge discovery. In: Miller, H.J., Han, J. (eds.) Geographic Data Mining and Discovery, pp. 53–73. Taylor and Francis, London, UK (2001)

Bédard, Y., Pageau, J., Caron, C.: Spatial Data Modeling: The Modul-R Formalism and CASE Technology. ISPRS Symposium, Washington DC, USA, August 1st–14th (1992)

Bédard, Y., Paquette, F.: Extending Entity/Relationship Formalism for Spatial Information Systems. AUTO-CARTO 9, 9th International Symposium on Automated Cartography, ASPRS-ACSM, Baltimore, USA, pp. 818–827, April 2nd–7th (1989)

Bédard, Y., Proulx, M.-J., Larrivée, S., Bernier, E.: Modeling Multiple Representations into Spatial Data Warehouses: A UML-based Approach. ISPRS WG IV/3, Ottawa, Canada, July 8th–12th, p. 7 (2002)

Bédard, Y.: Visual Modeling of Spatial Database, Towards Spatial PVL and UML. Geomatica **53**(2), 169–185 (1999)

Bedient, P.B., Huber W.C., Vieux, B.E.: Hydrology and Floodplain Analysis, 4th edn. Prentice Hall, New Jersey, USA (2007)

Been, K., Daiches, E., Yap, C.: Dynamic map labeling. In: Proc. of IEEE Information Visualization (2006)

Beginning MapServer: Open Source GIS Development by Bill Kropla (Apress Publications, 2005). This book shows the use of OGC-standard services in MapServer, one of the popular development platforms for integrating mapping technology into Internet applications

Beighley, R.E., Melack, M., Dunne, T.: Impacts of California's climatic regimes and coastal land use change on streamflow characteristics. J. Am. Water Resour. Assoc. **39**(6), 1419–1433 (2003)

Bell, B., Feiner, S.: Dynamic space management for user interfaces. In: Proc. UIST '00, pp. 239–248 (2000)

Bell, B., Feiner, S., Hoellerer, T.: View management for virtual and augmented reality. In: Proc. of UIST, pp. 101–110 (2001)

Bell, T., Witten, I.H., Cleary, J.G.: Modeling for text compression. ACM Computing Surveys. **21**, 557–591 (1989)

Belussi, A., Bertino, E., Catania, B., Damiani, M.L., Nucita, A: An authorization model for geographical maps. http://www.informatik.uni-trier.de/~ley/db/conf/gis/gis2004.html GIS 2004

Belussi, A., Bertino, E., Catania, B., Damiani, M.L., Nucita, A.: An authorization model for geographical maps. In: Proceedings of the12th ACM International Workshop on Geographic Information Systems, ACM-GIS, November 12–13, 2004, Washington, DC, USA, 2004

Ben-Akiva, M.: User's Guide for MITSIMLab and Road Network Editor (RNE), Massachusetts Institute of Technology, Intelligent Transportation Systems Program, November (2002)

Ben-Akvia, M., Bierlaire, M., Koutsopoulos, H., Mishalani, R.: DynaMIT: a simulation-based system for traffic prediction. In: DACCORD Short Term Forecasting Workshop. Delft, Netherlands (1998)

Benedikt, M., Dong, G., Libkin, L., Wong, L.: Relational expressive power of constraint query languages. J. ACM **45**(1), 1–34 (1998)

Benedikt, M., Keisler, H.J.: Definability over linear constraints. In: Clote, P., Schwichtenberg, H. (eds.) Proceedings of Computer Science Logic, 14th Annual Conference of the EACSL. Lecture Notes in Computer Science, vol. 1862, pp. 217–231, Springer-Verlag, Heidelberg (2000)

Benenson, I., Torrens, P. M.: Geosimulation: Automata-Based Modeling of Urban Phenomena. Wiley (2004)

Benetis, R., Jensen, C.S., Karciauskas, G., Saltenis, S.: Nearest neighbor and reverse nearest neighbor queries for moving objects. In: Proc. IDEAS, pp. 44–53 (2002)

Benetis, R., Jensen, C.S, Karčiauskas, G., Šaltenis, S.: Nearest and Reverse Nearest Neighbor Queries for Moving Objects. The VLDB Journal **15**(3), 229–249 (2006)

Benkert, M., Gudmundsson, J., Hübner, F., Wolle, T.: Reporting flock patterns. In: Proceedings of the 14th European Symposium on Algorithms. Lecture Notes in Computer Science, vol. 4168 pp. 660–671. Springer, Berlin, Heidelberg (2006)

Bennett, B.: Application of supervaluation semantics to vaguely defined spatial concepts. In: Montello, D.R. (ed.) Spatial Information Theory: Foundations of Geographic Information Science; Proceedings of COSIT'01, volume 2205 of LNCS, pp. 108–123. Springer, Morro Bay (2001)

Bennett, B.: What is a Forest? On the vagueness of certain geographic concepts. Topoi **20**, 189–201 (2001)

Bennett, F., Richardson, T., Harter, A.: Teleporting - making applications mobile. In: IEEE Workshop on Mobile Computing Systems and Applications, pp. 82–84. Santa Cruz, California (1994)

Benson, J.: LBS technology delivers information where and when its needed, Bus. Geogr. **9**(2), 20–22 (2001)

Bentley, J.L.: Multidimensional binary search trees used for associative searching. Communications of the ACM **18**(9):509–517 (1975)

Bentley, J.L.: Multidimensional Divide-and-Conquer. Communs. ACM **23**(4), 214–229 (1980)

Bentley, J.L., Saxe, N.B.: Decomposable Searching Problems I: Static-to-Dynamic Transformations. J. Algorithms **1**(4), 301–358 (1980)

Berchtold, B., Keim, D., Kriegel, H.-P.: Using extended feature objects for partial similarity retrieval. The VLDB Journal **6**(4), 333–348 (1997)

Berchtold, S., Böhm, C., Keim, D.A., Kriegel, H.: A cost model for nearest neighbor search in high-dimensional data space. In: Proceedings of ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), Tucson, Arizona, pp. 78–86 (1997)

Berchtold, S., Böhm, C., Kriegel, H-P.: The pyramid-technique: towards breaking the curse of dimensionality, pp. 142–153, SIGMOD Conference, (1998)

Berchtold, S., Böhm, C., Kriegel, H.-P.: The pyramid-tree: Breaking the curse of dimensionality. In: Haas, L.M., Tiwary, A. (eds.) SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, June 2–4, 1998, Seattle, Washington, USA, pp. 142–153. ACM Press, New York (1998)

Berchtold, S., Keim, D., Kriegel, H.P.: The X-tree: An index structure for high-dimensional data. In: Proc. 22th International Conference on Very Large Databases (VLDB'96), pp. 28–39. Morgan Kaufmann, San Francisco, CA, USA (1996)

Beresford, A.R.: Location privacy in ubiquitous computing. In: Technical Report No. 612 UCAM-CL-TR-612, University of Cambridge, Computer Laboratory, UK (2005)

Beresford, A.R., Stajano, F.: Location Privacy in Pervasive Computing. IEEE Pervasive Computing **2**(1):46–55 January–March (2003)

Beresford, A., Stajano, F.: Mix zones: User privacy in location-aware services. In: IEEE Workshop on Pervasive Computing and Communication Security (PerSec), pp. 127–131. IEEE Computer Society, Washington, DC, USA (2004)

Berggren, J.L., Jones A.: Ptolemy's Geography. Princeton University Press, Princeton (2000)

Berghel, H.: The client's side of the world wide web. Communications of the ACM **39**(1), 30–40 (1996)

Berg Insight, Press Release, Directions Magazine, November 15, (2006)

de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry: Algorithms and Applications. Springer-Verlag, Heidelberg (2000)

Berliner, L.M.: Hierarchical Bayesian time series models. In: Hanson, K., Silver, R. (eds.) Maximum Entropy and Bayesian Methods, pp. 15–22. Kluwer Academic Publishers, Dordrecht (1996)

Berman, F., Hey, A., Fox, G.: Grid Computing: Making The Global Infrastructure a Reality. Wiley, Indianapolis, IN, USA (2003)

Bernardinelli, L., Clayton, D.G., Pascutto, C., Montomoli, C., Ghislandi, M., Songini, M.: Bayesian analysis of space-time variation in disease risk. Stat. Med. **14**, 2433–2443 (1995)

Bernard, L., et al.: Towards an SDI Research Agenda. 11 EC-GIS Workshop. Alghero, Sardinia (2005)

Berndt, D., Clifford, J.: Using dynamic time warping to find patterns in time series. In AAAI Workshop on Knowledge Discovery in Databases, pp. 229–248 (1994)

Berners-Lee, T.: (1998) The Semantic Web Road Map. World Wide Web Consortium W3C Draft. http://www.w3.org/DesignIssues/Semantic

Berners-Lee, T. (2000) Semantic Web on XML. In: XML 2000. Available via. http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide1-0.html. Accessed 21 March 2007

Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web: a new form of web content that is meaningful to computers will unleash a revolution of new possibilities. Sci. Am. **284**, 34–43 (2001)

Bernhardsen, T.: Geographic information systems: an introduction, 3rd ed. John Wiley & Sons, New York (2002)

Bernstein, P.A., Giunchiglia, F., Kementsietsidis, A., Mylopoulos, J., Serafini, L., Zaihrayeu, I.: Data management for peer-to-peer computing: a vision. In: International Workshop on the Web and Databases (WebDB) 2002, June 6–7, Madison, WI

Berre, A.J.: Geospatial Services. ISO/TC211 WG4 No. 43 (1996)

Berry, B.J.L., Marble, D.F. (eds.): Spatial Analysis: A Reader in Statistical Geography. Prentice Hall, Englewood Cliffs, NJ (1968)

Berterretche, M., Hudak, A.T., Cohen, W.B., Maiersperger, T.K., Gower, S.T., Dungan, J.: Comparison of regression and geostatistical methods for mapping Leaf Area Index (LAI) with Landsat ETM+ data over a boreal forest. Remote Sens. Environ. **96**, 49–61 (2005)

Bertin, J.: Semiology Graphique. Mouton, The Hague (1967)

Bertino, E., Catania, B., Damiani, M.L., Perlasca, P.: GEO-RBAC: a spatially aware RBAC. In: Proceedings of the Tenth Symposium of ACM Access Control Technologies, SACMAT '05, Scandic Hasselbacken, Stockholm, June 1–3, 2005

Bertino, E., Damiani, M.L., Momini, D.: An access control system for a web map management service. In: Proceedings of the 14th International Workshop on Research Issues on Data Engineering, RIDE '04, Boston, USA, March 28–29, 2004

Bertolotto, M., Egenhofer, M.J.: Progressive transmission of vector map data over the World Wide Web. GeoInformatica **5**(4), 345–373 (2001)

Bertolotto, M., Egenhofer, M.J.: Progressive vector transmission. In: Proceedings of the 7th International Symposium on Advances in GIS, Kansas City, Nov 1999

Besag J., Newell, J.: The detection of clusters in rare diseases. J. R. Stat. Soc. Series A **154**, 143–155 (1991)

Besag, J.: Spatial interactions and the statistical analysis of lattice systems (with discussion). J. Royal Stat. Soc., Series B **36**, 192–236 (1974)

Besag, J., York, J.C., Mollie, A.: Bayesian image restoration, with two applications in spatial statistics (with discussion) Ann. Inst. Stat. Math. **43**, 1–59 (1991)

Bestavros, A.: WWW traffic reduction and load balancing through server-based caching. IEEE Concurrency: Spec. Issue Parallel Distrib. Technol. **5**, pp. 56–67 (1997)

Best Practices for Managing Geospatial Data (http://images.autodesk.com/adsk/files/%7B574931BD-8C29-4A18-B77C-A60691A06A11%7D_Best_Practices.pdf)

Bettini, C., Mascetti, S., Wang, X.S., Jajodia, S.: Anonymity in Location-based Services: Towards a General Framework. In: Proc. of the 8th International Conference on Mobile Data Management (MDM). IEEE Computer Society (2007)

Bettini, C., Wang, X.S., Jajodia, S.: Protecting privacy against location-based personal identification. In: Jonker, W., Petkovic, M. (eds.) Secure Data Management, Lecture Notes in Computer Science, vol. 3674, pp. 185–199. Springer (2005)

Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is 'Nearest Neighbor' Meaningful? Lecture Notes in Computer Science (1999)

Bhaduri, B., Bright, E., Coleman, P., Dobson, J.: LandScan: locating people is what matters. Geoinformatics **5**, 34–37 (2002)

Bhandarkar, S.M., Koh, J., Suk, M.: Multi scale image segmentation using a hierarchical self-organizing map. Neurocomputing **14**, 241–272 (1997)

Bhatti, A.U., Mulla, D.J., Frazier, B.E.: Estimation of soil properties and wheat yields on complex eroded hills using geostatistics

and thematic mapper images. Remote Sensing Environ. **37**(3), 181–191 (1991)
Bially, T.: Space-filling curves. Their generation and their application to bandwidth reduction. IEEE Trans. on Information Theory **15**(6), 658–664 (1969)
Bibel, B.: Arrest Locations as a Means for Directing Resources. In: Levine, N. (ed.) CrimeStat III: A Spatial Statistics Program for the Analysis of Crime Incident Locations (version 3.0), Chapter 6. Ned Levine & Associates and the National Institute of Justice, Houston, TX, Washington, DC (2004)
Biggs, N.L., Lloyd, E.K., Wilson, R.J.: Graph Theory 1736–1936. Oxford University Press, Oxford (1976)
Biggs, N.L., Lloyd, E.K., Wilson, R.J.: Graph Theory. Oxford University Press, Oxford (1998)
Bimonte, S., Tchounikine, A., Miquel, M. Towards a spatial multidimensional model. In Proceedings of the 8th ACM International Workshop on Data Warehousing and OLAP, November 4–5, pp. 39–46. Bremen, Germany (2005)
Birk R., Frederick, M., Dewayne, L.C., Lapenta, M.W.: NASA's Applied Sciences Program: transforming research results into operational success. Earth Imaging J. **3**(3), 18–23 (2006)
Bishr, Y.: Overcoming the semantic and other barriers to GIS interoperability. Int. J. Geogr. Inf. Sci. **12**(4), 299–314 (1998)
Bishr, Y.: Semantic Aspects of GIS. ITC, (1997)
Bithell, M., Macmillan, W.D.: Escape from the cell: Spatially explicit modeling with and without grids. Ecol. Modelling **200**, 59–78 (2007)
Bittner, T.: Approximate qualitative temporal reasoning. Annals of Mathematics and Artificial Intelligence **35**(1–2), 39–80 (2002)
Bittner, T.: On ontology and epistemology of rough location. In: Freksa, C., Mark, D.M. (eds.) Spatial Information Theory – Cognitive and Computational Foundations of Geographic Information Science, number 1661 in Lecture Notes in Computer Science, pp. 433–448. Springer, Proceedings of COSIT'99 (1999)
Bittner, T., Stell, J.G.: Approximate qualitative spatial reasoning. Spatial Cognition and Computation **2**(4), 435–466 (2002)
Bittner, T., Stell, J.G.: Rough sets in qualitative spatial reasoning. In: Rough Sets and Current Trends in Computing, Banff, October 2000, Proceedings, volume 2005 of Lecture Notes in Computer Science, pp. 445–453. Springer, Heidelberg (2001)
Bittner, T., Stell, J.G.: Stratified rough sets and vagueness. In: Kuhn, W., Worboys, M., Impf, S. (eds.) Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science. International Conference COSIT'03, pp. 286–303. Springer, Berlin (2003)
Bittner, T., Stell, J.G.: Vagueness and rough location. GeoInformatica **6**, 99–121 (2002)
Bivand, R.S.: Integrating grass 5.0 and r: GIS and modern statistics for data analysis. 7th Scandinavian Research Conference on Geographical Information Science. Aalborg, Denmark (1999)
Bivand, R.S., Neteler, M.: Open source geocomputation: using the r data analysis language integrated with grass GIS and postgresql data base systems (2000)
Bivand, R.S.: Using the r statistical data analysis language on grass 5.0 GIS database files. Comp. & Geosci. **26**, 1043–1052 (2000)
Black, P.E. Quadtree. In: Dictionary of Algorithms and Data Structures. NIST Available via: http://www.nist.gov/dads/HTML/quadtree.html (2006)
Blaer, P.: Robot path planning using generalized voronoi diagrams. http://www.cs.columbia.edu/~pblaer/projects/path_planner/
Block, C.R.: STAC hot spot areas: a statistical tool for law enforcement decisions. In: Proceedings of the Workshop on Crime Analysis Through Computer Mapping. Criminal Justice Information Authority, Chicago, IL (1994)
Bluestein, G.:, Companies track gridlock via cell phones. http://news.yahoo.com/s/ap/20061105/ap_on_hi_te/tracking_traffic;_ylt=AoQQdZOHq.vxPTOcxWTRCaDMWM0F;_ylu=X3oDMTA3ODdxdHBhBHNlYwM5NjQ-, Associated Press: Retreived 11/05/2006 (2006)
Blumberg, A.F., Mellor, G.L.: A description of a three-dimensional coastal ocean circulation model. In: Heaps, N.S. (ed.) Three-Dimensional Coastal Ocean Models, pp. 1–16. American Geophysical Union, Washington, D.C. (1987)
Blumberg, A.J., Keeler, L., Shelat, A.: Automated traffic enforcement which respects driver privacy. In: Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems, Vienna, Austria, September 2005, pp. 650–655. IEEE Computer Society, Washington, DC, USA (2005)
Bochnak, J., Coste, M., Roy, M.-F.: Géométrie algébrique réelle. Springer-Verlag, Berlin/Heidelberg (1987)
Bogaert, P.: Comparison of kriging techniques in a space-time context. Math. Geol. **28**, 73–86 (1996)
Bogaert, P., D'Or, D.: Estimating soil properties from thematic soil maps-The BME approach. Soil Sci. Soc. Am. J. **66**, 1492–1500 (2002)
Bogaert, P.: Spatial prediction of categorical variables: the BME approach. Stochastic Environ. Res. Risk Assess. **16**, 425–448 (2002)
Boguna, M., Pastor-Satorras, R., Vespignani, A.: Epidemic Spreading in Complex Networks with Degree Correlations. In: Pastor-Satorras, R., Rubi, M., Diaz-Guilera, A. (eds.) Statistical Mechanics of Complex Networks. Lect. Notes Phys. **625**, 127–147 (2003)
Böhlen, M.H., Jensen, C.S.: Temporal data model and query language concepts. Encyclopedia of Information Systems, 4, Academic Press, Inc. (2003)
Böhm, C., Berchtold, S., Keim, D.: Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. ACM Comput. Surv. **33**(3):322–373 (2001)
Bohm, C., Klump, G., Kriegel, H.-P.: xz-ordering: A space-filling curve for objects with spatial extension. In: Proc. of 6th Intl. Symp. on Large Spatial Databases, SSD, Hong Kong, July 1999, pp. 75–90
Bohn, J., Vogt, H.: Robust Probabilistic Positioning Based on High-Level Sensor Fusion and Map Knowledge. Tech. Rep. 421. Institute for Pervasive Computing, Distributed Systems Group, Swiss Federal Institute of Technology (ETH), Zurich (2003)
Bollobás, B., Das, G., Gunopulos, D., Mannila, H.: Time-series similarity problems and well-separated geometric sets. In Proc. of SCG (1997)
Bollobás, B.: Graph Theory: An Introductory Course, p.~12. Springer, Berlin (1979)
Bonczek, R.H., Holsapple, C.W., Whinston, A.B.: Foundations of Decision Support Systems. Academic Press, Orlando, FL (1981)
Bonet, J.S.D.: Image Reprocessing for Rapid Selection in "Pay attention mode". MIT Press, Boston, MA (2000)
Bonifati, A., Ceri, S.: Comparative Analysis of Five XML Query Languages, ACM SIGMOD Record, vol. 29, pp. 68–79, March, (2000)
Bonnet, P., Gehrke, J., Seshardi, P.: Toward sensor database systems. Paper presented at the 2nd International Conference on Mobile Data Management (MDM 2001), pp. 3–14. Hong Kong, China, January (2001)

Booch, G., Jacobson, I., Rumbaugh, J.: The Unified Modeling Language User Guide. Addison-Wesley, Reading, MA (1998)

Borchert, A.: Multimedia atlas concepts. In: Cartwright, W., Peterson, M., Gartner, G. (eds.) Multimedia Cartography, pp. 75–86. Springer, Berlin Heidelberg New York (1999)

Bordegoni, M., Faconti, G., Feiner, S., Maybury, M., Rist, T., Ruggieri, R., Trahanias, P., Wilson, M.: A standard reference model for intelligent multimedia presentation systems. Comp. Stand. Interf. **18**(6–7), 477–496 (1997)

Borges, K.A., Davis, C.D., Laender, A.H.F.: OMT-G: An object-oriented data model for geographic applications. GeoInformatica **5**, 221–260 (2001)

Born, M., Wolf E.: Principles of optics: electromagnetic theory of propagation, interference and diffraction of light, 6th edn. Pergamon Press, Oxford, New York (1980)

Borràs, J., Hatrack, P., Mandayam, N.B.: Decision theoretic framework for NLOS identification. In: IEEE VTC 1998, **2**, 1583–1587 (1998)

Borrmann, A., van Treeck, C., Rank, E.: Towards a 3D Spatial Query Language for Building Information Models. In: Proceedings of the 11th Int. Conf. on Computing in Civil and Building Engineering (ICCCBE-XI), Montreal, Canada (2006)

Borsuk, M.E., Reckhow, K.H.: Summary Description of the Neuse Estuary Bayesian Ecological Response Network (Neu-BERN). (2000) http://www2.ncsu.edu/ncsu/CIL/WRRI/neuseltm.html (Dec. 26, 2001)

Boscoe, F.P., Ward, M.H., Reynolds, P.: Current practices in spatial analysis of cancer data: data characteristics and data sources for geographic studies of cancer. Int. J. Health Geogr. **3**, article 28 (2004)

Bose, P., Morin, P., Stojmenovic, I., Urrutia, J.: Routing with guaranteed delivery in ad hoc wireless networks. Wireless Networks **7**, 609–616 (2001)

Bosowski, E.F., Feeman T.G.: The use of scale factors in map analysis: an elementary approach. Cartographica **34**(4), 35–44 (1997)

Bossler, J.D., Jensen, J.R., McMaster, R.B., Rizos, C. eds.: Manual of Geospatial Science and Technology. London: Taylor and Francis, 2002

Botts, M. (eds.) (2005) OpenGIS® Sensor Model Language (SensorML)Implementation Specification, OGC doi:05-086 http://portal.opengeospatial.org/files/?artifact_id=12606

Bourassa, S., Hoesli, M., Peng, V.: Do housing submarkets really matter? J. Housing Econ. **12**(1), 12–28 (2003)

Bowden, L.: The story of IFF (identification friend or foe), IEEE Proceedings, Part A: Physical Science, Measurement and Instrumentation, Management and Education **132**(6), 435–437 (1985)

Boxall, J.: Geolibraries, the Global Spatial Data Infrastructure and Digital Earth. Int. J. Spec. Libr. **36**(1), 1–21 (2002)

Box, G., Jenkins, G., Reinsel, G.: Time Series Analysis: Forecasting and Control. Prentice Hall, Upper Saddle River, NJ (1994)

Braga, A.A., Weisburd, D.L., et al.: Problem-oriented policing in violent crime places: A randomized controlled experiment. Criminology **7**, 541–580 (1999)

Brail, R.K., Klosterman, R.E. (eds.): Planning Support Systems: integrating geographic information systems, models, and visualization tools. ESRI Press (2001)

Braitenberg, V.: Vehicles, experiments in synthetic psychology. MIT Press, Cambridge, MA (1984)

Brakatsoulas, S., Pfoser, D., Sallas, R., Wenk, C.: On map-matching vehicle tracking data. In: Proc. 31st VLDB conference, pp. 853–864 (2005)

Brakatsoulas, S., Pfoser, D., Theodoridis, Y: Revisiting r-tree construction principles. In: Proc. of the East European Conference on Advances in Databases and Information Systems, Bratislava, Slovakia, 8–11 Sept 2002, pp. 149–162

Brandeis, L., Warren, S.: The right to privacy. Harvard Law Rev. **4**(5), 193–220 (1890)

Brandt, R., Groenewoudt, B.J., Kvamme, K.L.: An experiment in archaeological site location: Modeling in the netherlands using GIS techniques. World Arch. **24**(2), 268–282 (1992)

Brantingham, P., Brantingham, P.: Environmental Criminology. Waverland Press, Prospect Heights, Il (1981)

Brantingham, P., Brantingham, P.: Location quotients and crime hotspots in the city. In: Block, C., Dabdoub, M., Fregly, S. (eds.) Crime analysis through computer mapping. Police Executive Research Forum, Washington DC (1995)

Brantingham, P.L., Brantingham, P.J.: Notes on the geometry of crime. In: Brantingham, P.J., Brantingham, P.L. (eds.) Environmental Criminology, pp. 27–54. Waveland Press Inc., Prospect Heights, IL (1981)

Brasington, D., Hite, D.: Demand for environmental quality: a spatial hedonic analysis. Reg. Sci. Urban Econ. **35**(1), 57–82 (2005)

Brassel, K.E., Weibel, R.: A review and conceptual framework of automated map generalization. International Journal of Geographical Information Systems. **2** 229–244 (1988)

Braun, J.: Dempster-Shafer Theory and Bayesian Reasoning in Multisensor Data FuSion. Sensor Fusion: Architectures, Algorithms and Applications IV. In: Proceedings of SPIE 4051, pp. 255–266 (2000)

Bray, C., Woodsford, P.: Positional Accuracy Improvement: Impacts of improving the positional accuracy of GI databases. In: EuroSDR official workshop publication. no. 48, Dublin. http://www.eurosdr.net/km_pub/no48/html/positional/positional_index.htm (2004)

Breinholt, G., Schierz, C.: Algorithm 781: Generating hilbert's space-filling curve by recursion. ACM Trans. Math. Softw., TOMS **24**(2),184–189 (1998)

Brenner, C., Sester, M.: Continuous generalization for small mobile displays. In: Agouris, P., Croitoru, A. (eds.) Next Generation Geospatial Information, pp. 33–41. Taylor & Francis (2005)

Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: Lof: Identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, p. 93–104, Dallas, Texas, United States, 14–19 May 2000

Brewer, C.: Designing Better Maps. ESRI Press (2005)

Brickley, D. (2003) Basic Geo (WGS84 lat/long) Vocabulary. http://www.w3.org/2003/01/geo/. Accessed 21 March 2007

Brillinger, D.R., Preisler, H.K., Ager, A.A., Kie, J.G.: An exploratory data analysis (EDA) of the paths of moving animals. J. Stat. Plan. Inf. **122**(2), 43–63 (2004)

Brinkhoff, T., Kriegel, H.P., Seeger, B.: Efficient processing of spatial joins using r-trees. In: Proc. of ACM SIGMOD Int'l Conference (1993)

Brinkoff, T., Kreigel, H.-P., Schneider, R., Seegar, B.: Multi-step processing of spatial joins. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, Minneapolis, MN, 24–27 May 1994

The British Broadcasting Corporation: Stopping the book thieves. http://news.bbc.co.uk/2/hi/uk_news/1958424.stm, The BBC News: Retreived 10/10/2006 (2002)

Brodeur, J., Bédard, Y., Edwards, G., Moulin, B.: Revisiting the concept of geospatial data interoperability within the scope of a human communication process. Trans. GIS **7**, 243–265 (2003)

Brodeur, J., Bédard, Y., Proulx, M.J.: Modeling geospatial application databases using UML-based repositories aligned with international standards in geomatics. In: Proceedings of the 8th ACM Symposium on Advances in Geographic Information Systems (ACMGIS), Washington, D.C., USA, 6–11 Nov 2000

Brodie, M., Mylopoulos, J., Schmidt, J.: On Conceptual Modelling. Topics in Information Systems. Springer-Verlag, New York/Heidelberg (1984)

Brodsky, A., Segal, V., Chen, J., Exarkhopoulo, P.: The CCUBE constraint object-oriented database system. Constraints **2**(3–4), 245–277 (1997)

Brooks, A., Makarenko, A., Kaupp, T., Williams, S., Durrant-Whyte, H.: Implementation of an indoor active sensor network. In: Int. Symp. on Exp. Robotics, Singapore (2004)

Browne, D., Totterdell, P., Norman, M. (eds.): Adaptive User Interfaces. Computers and People Series. Academic Press, London (1990)

Brown, G.G., Valliliou, A.L.: Optimizing Disaster Relief: Real-Time Operational and Tactical Decision Support. Naval Res. Logistics **40**, 1–23 (1993)

Brown, J.D., Heuvelink, G.B.M.: The Data Uncertainty Engine (DUE): a software tool for assessing and simulating uncertain environmental variables. Comput. & Geosci. **33**, 172–190 (2007)

Brownstein, J.S., Holford, T.R., Fish, D.: A climate-based model predicts the spatial distribution of the Lyme disease vector *Ixodes scapularis* in the United States. Envrion. Health Persp. **111**, 1152–1157 (2003)

Bruce, J. (1993) Review of tenure terminology. Land Tenure Center Report. http://agecon.lib.umn.edu/ltc/ltctb01.pdf. Accessed 14 Aug 2007

De Bruin, S., Bregt, A., Van De Ven, M.: Assessing Fitness for Use: The Expected Value of Spatial Data Sets. International Journal of Geographical Information Sciences **15**(5), 457–471 (2001)

de Bruin, S.: Querying probablistic land cover data using fuzzy set theory. Int. J. Geogr. Inf. Sci. **14**(4), 359–372 (2000)

Brumberger, H., Goodisma, J.: Voronoi cells: An interesting and potentially useful cell model for interpreting the small-angle scattering of catalysts. In: Journal of Applied Crystallography, vol. 16, pp. 83–88. (1983)

Brunato, M., Battiti, R.: Statistical learning theory for location finngerprinting in wireless lans. Computer Networks **47**(6), 825–845 (2005)

Bruns, D.A.: Land use planning with CITYgreen: GIS software evaluates development impacts on "green" infrastructure. Land Bulletin of the National Consortium for Rural Geospatial Innovations. University of Wisconson, Madison, WI. www.ruralgis.org In Press (2007)

Brunsdon, C., Fotheringham, A.S., Charlton, M.: Geographically weighted regression: a method for exploring spatial nonstationarity. Geogr. Anal. **28**, 281–98 (1996)

Bruns, D.A., Sweet, T.: Geospatial tools to support watershed environmental monitoring and reclamation: assessing mining impacts on the Upper Susquehanna-Lackawanna American Heritage River. U.S. Department of Interior, Office of Surface Mining. Conference Proceedings, Advanced Integration of Geospatial Technologies in Mining and Reclamation, December 7–9, Atlanta, GA. http://www.tips.osmre.gov/GeoSpatial/post-conference/Papers/Bruns_Sweet1_PA.pdf

Bruns, D.A., Sweet, T.: PPGIS in Rural America: Integration of Geospatial Technologies and Resource Leveraging in Data Acquisition for an American Heritage River community. In: Proceedings of the Annual PPGIS Conference of the Urban and Regional Information Systems Association (URISA), pp. 43–70. University of Wisconsin, Madison, WI (2004)

Bruns, D.A., Sweet, T., Toothill, B.: Upper Susquehanna-Lackawanna River Watershed, Section 206, Ecosystem Restoration Report. Phase I GIS Environmental Master Plan. Final Report to U.S. Army Corps of Engineers, Baltimore District, MD (2001)

Bruns, D.A., Wiersma, G.B.: Design of environmental monitoring systems: a geospatial perspective. In: Wiersma, G.B. (ed.) Environmental Monitoring, pp. 1–35. Lewis Publishers (2004)

Bruns, H., Eggenhofer, M.: Similarity of Spatial Scenes. In: Molenaar, M., Kraak, M.J. (eds.) Proc. 7th Int. Symposium on Spatial Data Handling, pp. 31–42. Taylor and Francis, London (1996)

Brusilovsky, P.: Adaptive Hypermedia. User Modeling User-Adapted Interaction **11**, 87–110 (2001)

BSCS Bioinformatics and the human genome project: A Curriculum Supplement for High School Biology. Developed by BSCS under a contract from the Department of Energy (2003)

Buber, M.: Ich und Du. Reclam jun. GmbH & Co., Stuttgart (1995)

Buchin, K., Sousa, M.C., Döllner, J., Samavati F., Walther, M.: Illustrating terrain using direction of slope and lighting. In: 4th ICA Mountain Cartography Workshop (2004)

Bugayevskiy, L.M., Snyder J.P.: Map Projections: A Reference Manual. Taylor and Francis, London (1995)

Bulusu, N., Heidemann, J., Estrin, D.: GPS-less low cost outdoor localization for very small devices. IEEE Personal Comm. **5**(5), 28–34 (2000)

Bundy, G., Jones, C., Furse, E.: Holistic generalization of large-scale cartographic data. [20], 106–119

Buneman, P., Davidson, S.B., Kosky, A.: Theoretical aspects of schema merging. In: International Conference on Extending Database Technology (EDBT) 1992, March 23–27, Vienna, Austria

Bunge, W.: Theoretical Geography. Gleerup, Lund (1966)

Buragohain, C., Gandhi, S., Hershberger, J., Suri, S.: Contour approximation in sensor networks. In: DCOSS (2006)

Burghardt, D., Neun, M., Weibel, R.: Generalization services on the web—classification and an initial prototype implementation. Cartogr. Geogr. Inf. Sci. **32**, 257–268 (2005)

Burghardt, D., Purves, R. S., Edwardes, A.J.: Techniques for on-the-fly generalisation of thematic point data using hierarchical data structures. In: Proceedings of the GIS Research UK 12th Annual Conference, Norwich, 28–30 April (2004)

Burl, M.C.: Mining for image content. In Systemics, Cybernetics, and Informatics / Information Systems: Analysis and Synthesis, Orlando, FL (July 1999)

Burnham, K.P., Anderson, D.R.: Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach, 2nd edn. Springer, New York (2002)

Burns, L.D.: Transportation, Temporal and Spatial Components of Accessibility. Lexington Books (1979)

Burrough, P., Frank, A.M. (eds.): Specialist Meeting on Geographical Objects with Undetermined Boundaries, GISDATA. Francis Taylor (1996)

Burrough, P., Frank, A.U. (eds.): Geographic Objects with Indeterminate Boundaries, GISDATA Series II. Taylor and Francis, London (1995)

Burrough, P.A.: Fuzzy mathematical methods for soil survey and land evaluation. J. Soil Sci. **40**, 477–492 (1989)

Burrough, P.A., McDonnell, R.A.: Principles of geographical information systems. Oxford University Press, Oxford (1998)

Burrough, P.A.: Natural Objects with Indeterminate Boundaries. In: Burrough, P.A., Frank, A.U. (eds.) Geographic Objects with Indeterminate Boundaries, pp. 345. Taylor & Francis, Ltd., London (1996)

Burrough, P.A., van Gaans, P.F.M., Hootsmans, R.: Continuous classification in soil survey: spatial correlation, confusion and boundaries. Geoderma, **77**, 115–135 (1997)

Bursik, R.J., Grasmick, H.G.: Neighborhoods and Crime: The Dimensions of Effective Community Control. Lexington Books, New York, NY (1993)

Burton, F., Kollias, J., Kollias, V., Matsakis, D.: Implementation of overlapping B-trees for time and space efficient representation of collection of similar files. Comput. J., **33**(3), 279–280 (1990)

Burton, M.L., Hicks, M.J.: Hurricane Katrina: Preliminary Estimates of Commercial and Public Sector Damages. Center for Business and Economic Research. Marshall University, Huntington (2005)

Bustos, B., Keim, D., Saupe, D., Schreck, T., Vranić, D.: Feature-based similarity search in 3D object databases. ACM Computing Surveys **37**(4), 345–387 (2005)

Buttenfield, B., Gahegan, M., Miller, H. Geospatial Data Mining and Knowledge Discovery. http://www.ucgis.org/priorities/research/research_white/2000%20Papers/emerging/gkd.pdf

Buttenfield, B.P.: Looking Forward: Geographic Information Services and Libraries in the Future. Cartography and Geographic Information Systems, **25**(3), 161–173

Buttenfield, B.P., McMaster, R.B.: Map Generalization: Making Rules for Knowledge Representation. Longman, London (1991)

Buttenfield, B.P.: Transmitting vector geospatial data across the Internet. In: Proceedings of the Second International Conference on Geographic Information Science, 25–28 September 2002, Boulder, Colorado, USA. Lecture notes in Computer Science 2478, pp. 51–64. Springer, Berlin (2002)

Butz, A.R.: Alternative Algorithm for Hilbert's Space-Filling Curve. IEEE Trans. Comput. **C-20**(4), 424–426 (1971)

Buyukokkten, O., Cho, J., Garcia-Molina, H., Gravano, L., Shivakumar, N.: Exploiting geographical location information of web pages. In: ACM SIGMOD Workshop on The Web and Databases, Philadelphia, Pennsylvania, USA (1999)

Buziek, G.: Eine Konzeption der kartographischen Visualisierung. Habilitation thesis, University of Hannover (2002)

Cacuci, D.G.: Sensitivity and Uncertainty Analysis, Vol. 1: Theory. Chapman and Hall, Boca Raton, FL (2003)

CAD and GIS – Critical Tools, Critical Links: Removing Obstacles Between CAD and GIS Professionals (http://images.autodesk.com/adsk/files/3582317_CriticalTools0.pdf)

Cai, G., Bolelli, L., MacEachren, A.M., Sharma, R., Fuhrmann, S., McNeese, M.: Geocollaborative crisis management: using maps to mediate eoc–mobile team collaboration. In: dg.o '04: Proceedings of the 2004 annual national conference on Digital government research, pp. 1–2. Digital Government Research Center (2004)

Cai, G.R., MacEachren, A.M., Brewer, I., McNeese, M., Sharma, R., Fuhrmann, S.: Map-mediated GeoCollaborative Crisis Management. In: Intelligence and Security Informatics, Proceedings, vol. 3495. pp. 429–435 (2005)

Cai, M., Keshwani, D., Revesz, P.Z.: Parametric Rectangles: A Model for Querying and Animation of Spatiotemporal Databases. In: Proceedings of the 7th International Conference on Extending Database Technology, pp. 430–444. Konstanz, Germany (2000)

Cai, M., Revesz, P.: Parametric R-Tree: An index structure for moving objects. 10th International Conference on Management of Data, pp. 57–64. McGraw-Hill, New Dehli (2000)

Cai, Q., Rushton, G., Bhaduri, B., Bright, E., Coleman, P.: Estimating small-area populations by age and sex using spatial interpolation and statistical inference methods. Trans. GIS **10**, 577–598 (2006)

Caldwell, T.: On Finding minimum Routes in a Network with Turn Penalties. Commun. ACM **4**(2), 107–108 (1961)

Calvanese, D., Giacomo, G.D., Lenzerini, M., Nardi, D., Rosati: Schema and Data Integration Methodology for DWQ. Technical report, DWQ Consortium. TR# DWQ-UNIROMA-004, Sept. 1998. Università di Roma "La Sapienza", Roma (1998)

Câmara, G., Fonseca, F., Monteiro, A.M., Onsrud, H.: Networks of innovation and the establishment of a spatial data infrastructure in Brazil. Inf. Technol. Dev. **12**, 255–272 (2006)

Cameron, A.C., Trivedi, P.K.: Regression Analysis of Count Data. Cambridge University Press, Cambridge, UK (1998)

Campbell, J.B.: Introduction to Remote Sensing, 4th edn. p. 626. The Guilford Press, New York (2006)

Canada Centre for Remote Sensing. http://ccrs.nrcan.gc.ca/index_e.php. Accessed November 2006

Canter, D., Gregory, A.: Identifying the residential location of rapists. J. Forens. Sci. Soc. **34**(3), 169–175 (1994)

Canter, D.: Mapping Murder: The Secrets of Geographic Profiling. Virgin Publishing, London (2003)

Cao, H., Cheung, D.W., Mamoulis, N.: Discovering partial periodic patterns in discrete data sequences. In: Proc. of PAKDD (2004)

Cao, H., Mamoulis, N., Heung, C.D.W.K.: Mining frequent spatio-temporal sequential patterns. In ICDM (2005)

Cao, H., Wolfson, O., Trajcevski, G.: Spatio-temporal data reduction with deterministic error bounds. Journal of Very Large Databases **15**(3) (2006)

Cao, P., Wang, Z.: Efficient top-k query calculation in distributed networks. In: PODC 04: Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing, pp. 206–215, St. John's, Newfoundland, Canada, 25–28 July 2004

Čapkun, S., Hamdi, M., Hubaux, J.P.: GPS-free positioning in mobile ad-hoc networks. In: 34th IEEE Hawaii Int. Conf. on System Sciences (HICSS-34), Jan. (2001)

Cardei, M., Wu, J.: Coverage in wireless sensor networks. In: Ilyas, M., Magboub, I. (eds.) Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems, pp. 19:1–19:10. CRC, Boca Raton, FL (2004)

Cardoso, J.: Semantic Web Services: Theory, Tools and Applications. Idea Group, (2007)

Cardoso, J., Sheth, A.: Semantic Web Services, Processes and Applications. Springer-Verlag, USA (2006)

Card, S.K., MacKinlay, J.D., Shneiderman, B.: Readings in Information Visualization: Using Vision to Think. Morgan Kaufmann, San Francisco (1999)

Carley, K., Fridsma, D., Casman, E., Altman, N., Chang, J., J., Kaminski, B., Nave, D., Yahja, A.A.: BioWar: Scalable Multi-Agent Social and Epidemiological Simulation of Bioterrorism Events. NAACSOS conference proceedings, Pittsburgh, PA (2003)

Carlin, B.P., Louis, T.A.: Bayes and Empirical Bayes Methods for Data Analysis. 2nd edn. Chapman and Hall/CRC Press, Boca Raton, FL (2000)

Carl, J.R., DeAngelo, M.M., Ruskey, D.C.: Sustainable development plan and runoff study for a proposed development in the Hick's Creek watershed. In: Montz, B.E., Tobin, G.A. (eds.) Papers and Proceedings of the Applied Geography Conferences, vol. 25, pp. 9–17. Binghamton, NY (2002)

Carsjens, G.J., van Lammeren, R.J.A., Ligtenberg, A.: STEPP: a strategic tool for integrating environmental aspects into planning procedures. In: Geertman, S., Stillwell, J. (eds.) Planning

Support Systems in Practice, pp. 139–154. Springer, New York, NY (2003)
Carstairs, V.: Small area analysis and health service research. Community Med. **3**, 131–139 (1981)
Carver, S.: Developing web-based GIS/MCE: Improving access to data and spatial decision support tools. In: Thill, J.C.(eds.) Multicriteria decision-making and analysis: A geographic information sciences approach, pp. 49–75. Ashgate Publishing Ltd., Aldershot, United Kingdom (1999)
Carver, S.J.: Open spatial decision making on the Internet. School of Geography, University of Leeds, Yorkshire, England (1996)
Casati, R., Smith, B., Varzi, A.C.: Ontological Tools for Geographic Representation. In: Guarino, N. (ed.) Formal Ontology in Information Systems, pp. 77–85. IOS Press, Amsterdam (1998)
Casati, R., Varzi, A. (eds.): Events. The International Research Library of Philosophy. Dartmouth, Aldershot, UK (1996)
Casati, R., Varzi A.: Fiat objects. Topoi, **20**,131–148 (2001)
Casati, R., Varzi, A.: The structure of spatial localization. Philos. Stud. **82**, 205–239 (1996)
Catlett, C., Smarr, L.: Metacomputing. Commun. ACM **35**, 44–52 (1992)
Catovic, A., Sahinoglu, Z.: The Cramér-Rao bounds of hybrid TOA/RSS and TDOA/RSS location estimation schemes. Mitsubishi Electric Research Lab, Tech. Rep. TR-2003-143, Jan. 2004. [Online]. http://www.merl.com/reports/docs/TR2003-143.pdf
Cau, P., Cadeddu, A., Gallo, C., Lecca, G., Marrocu, M.: Estimating Available Water Resources of the Sardinian Island Using the SWAT Model. SWAT2003 2nd International SWAT Conference Proceedings, Bari, Italy. Retrieved July 1, (2004) from SWAT Proceedings Web site: http://www.brc.tamus.edu/swat/2ndswatconf/2ndswatconfproceeding.pdf (64–70) (2003)
CC2431: System-on-Chip for 2.4 GHz ZigBee/IEEE 802.15.4 with Location Engine. http://www.ti.com/lit/gpn/cc2431 (2005)
Cecconi, A., Galanda, M.: Adaptive zooming in web cartography. Comput. Graphics Forum **21**, 787–799 (2002)
Cehver, V., McClellan, J.H.: Sensor array calibration via tracking with the extended Kalman filter. In: Proceedings of 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 5, pp. 2817–2820 (2001)
Center for Geospatial Intelligence at the University of Missouri – Columbia (UMC). http://geoint.missouri.edu/
Center for International Earth Science Information Network: Gridded Population of the World (GPWv3). Columbia University, New York. http://sedac.ciesin.org/gpw/ (2007). Accessed 17 Sept 2007
Centers for Disease Control and Prevention (CDC). http://www.cdc.gov/ncidod/dvbid/westnile. 11 Sept 2007
Cerioli, A., Riani, M.: The ordering of spatial data and the detection of multiple outliers. J. Comput. Graphical Stat. **8**(2):239–258 June 1999
Ceri, S., Pelegatti, G.: Distributed Databases: Principles and Systems, McGraw-Hill (1984)
Cervone, G., Singh, R.P., Kafatos, M., Yu, C.: Wavelet maxima curves of surface latent heat flux anomalies associated with Indian earthquakes. Nat. Hazards Earth Syst. Sci. **5**, 87–99 (2005)
Chabini, I.: Discrete Dynamic Shortest Path Problems in Transportation Applications. Trans Res Rec **1645**, 170–175 (1998)
Chainey, S., Ratcliffe, J.: GIS and Crime Mapping. John Wiley and Sons, Ltd., Hoboken, NJ (2005)
Chakhar, S., Martel, J.M.: Enhancing geographical information systems capabilities with multi-criteria evaluation functions. J. Geogr. Inf. Dec. Anal. **7**(2), 47–71 (2003)
Chakhar S., Mousseau, V.: DMA: An algebra for multicriteria spatial modeling. In: Proccedings ICA Workshop on Geospatial Analysis and Modeling, 8 July, pp. 155–185, Vienna, Austria (2006)
Chakka, V.P., Everspaugh, A., Patel, J.M.: Indexing Large Trajectory Data Sets With SETI. In: Biennial Conf. on Innovative Data Systems Research (CIDR), Asilomar, CA, USA 2003
Chakrabarti, C., Vishwanath, M., Owens, R.: Architectures for wavelet transforms: a Survey. J. VLSI Signal Process, pp. 171–192 (2005)
Chakrabarti, K., Garofalakis, M.N., Rastogi, R., Shim, K.: Approximate query processing using wavelets. In: VLDB '00: Proceedings of the 26th International Conference on Very Large Data Bases, pp. 111–122. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2000)
Chakrabarti, K., Keogh, E.J., Pazzani, M., Mehrotra, S.: Locally adaptive dimensionality reduction for indexing large time series databases. ACM Trans. Datab. Syst. **27**(2), 188–228 (2002)
Chakrabarti, K., Mehrotra, S.: Dynamic Granular Locking Approach to Phantom Protection in R-trees. In: Proceedings of IEEE International Conference on Data Engineering, Orlando, Florida, 23–27 Feb 1998
Chakrabarti, K., Mehrotra, S.: Efficient Concurrency Control in Multi-dimensional Access Methods. In: Proceedings of ACM SIGMOD International Conference on Management of Data, Philadelphia, Pennsylvania, 1–3 June 1999
Chakravorty, R., Agarwal, S., Banerjee, S., Pratt, I.: MoB: A mobile bazaar for wide-area wireless services, International Conference Mobile Computing and Networking (MobiCom'05), Aug 2005
Chakravorty, S.: Identifying crime clusters: The spatial principles. Middle States Geogr. **28**, 53–58 (1995)
Chance, A., Newell, R.G., Theriault, D.G.: An Object-Oriented GIS – Issues and Solutions, EGIS '90 Conference Proceedings, Amsterdam, April (1990)
Chandran, S., Gupta, A.K., Patgawkar, A.: A Fast Algorithm to Display Octrees. (2004) http://www.cse.iitb.ac.in/~sharat/papers/linearOctreee.pdf
Chandrasekaran, B., Johnson, T., Benjamins, V.: Ontologies: what are they? Why do we need them?. IEEE Intelligent Syst. Their Appl. **14**, 20–26 (1999)
Chan, F.K., and Fu, A.W.: Haar Wavelets for Efficient Similarity Search of Time-Series: With and Without Time Warping. IEEE Trans. on Knowl. Data Eng. **15**(3), 678–705 (2003)
Chang, H.-J.: Understanding the Relationship between Institutions and Economic Development: Some Key Theoretical Issues. UNU/WIDER Discussion Paper 2006/05. July 2006. http://www.wider.unu.edu/publications/dps/dps2006/dp2006-05.pdf. Accessed 14 Aug 2007
Chang, Y.-C., Bergman, L.D., Castelli, V., Li, C.-S., Lo, M.-L., Smith, J.R.: The onion technique: indexing for linear optimization queries. SIGMOD Conference 391–402 (2000)
Chan, L.M., Zeng, M.L.: Metadata Interoperability and Standardization – A Study of Methodology Part 1, D-Lib Magazine. **12**(6) (2006). ISSN 1082–9873. http://www.dlib.org/dlib/june06/chan/06chan.html
Chaplot, V., Saleh, A., Jaynes, D.B., Arnold, J.: Predicting water, sediment, and NO_3-N Loads Under Scenarios of land-use and management practices in a flat watershed. Water, Air, Soil Pollut. **154**(1–4), 271–293 (2004)
Charlton, E.F.: The Octree. (1997)http://hpcc.engin.umich.edu/CFD/users/charlton/Thesis/html/node29.html. University of Michigan, Ann Arbor, MI, USA

Chartrand, G.: The Königsberg Bridge Problem: An Introduction to Eulerian Graphs. In: Introductory Graph Theory. Dover, New York (1985)

Chaudhuri, S., Dayal, U.: An overview of data warehousing and OLAP technology. SIGMOD Record **26**, 65–74 (1997)

Chaum, D.: Security without identification: Transaction systems to make big brother obsolete. Commun. ACM **28**(10), 1030–1044 (1985)

Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM **24**(2):84–88 (1981)

Chavez, E., Figueroa, K., et al.: A Fast Algorithm for the All k Nearest Neighbors Problem in General Metric Spaces

Chazelle, B., Dobkin, D.: Decomposing a polygon into its convex parts. In: Proc. 11th Annual ACM Symposium on Theory of Computing, pp. 38–48., Atlanta, Georgia, 1979

Chen, B.-C., Chen, L., Lin, Y., Ramakrishnan, R.: Prediction cubes. In: Proc. 2005 Int. Conf. Very Large Data Bases (VLDB'05), pp. 982–993. Trondheim, Norway, Aug. (2005)

Chen, C.-C., Knoblock, C.A., Shahabi, C.: Automatically conflating road vector data with orthoimagery. Geoinformatica **10**(4), 495–530 (2006)

Chen, C.-C., Knoblock, C.A., Shahabi, C., Chiang, Y.-Y., Thakkar, S.: Automatically and Accurately Conflating Orthoimagery and Street Maps. In: Proceedings of the 12th ACM International Symposium on Advances in Geographic Information Systems. Washington, D.C., USA (2004)

Chen, C.-C., Shahabi, C., Knoblock, C.A., Kolahdouzan, M.: Automatically and Efficiently Matching Road Networks with Spatial Attributes in Unknown Geometry Systems. In: Proceedings of the Third Workshop on Spatio-temporal Database Management (colocated with VLDB2006). Seoul, Korea (2006)

Chen, C.-C., Shahabi, C., Knoblock, C.A.: Utilizing Road Network Data for Automatic Identification of Road Intersections from High Resolution Color Orthoimagery. In: Proceedings of the Second Workshop on Spatio-temporal Database Management (colocated with VLDB2004). Toronto, Canada (2004)

Chen, C.X., Wang, H., Zaniolo, C.: Toward Extensible Spatiotemporal Databases: an approach based on User-Defined Aggregates. In: de Caluwe, R., et al.(eds.) Spatio-temporal Databases: Flexible Querying And Reasoning, pp. 55–74. Springer, Berlin, Heidelberg, New York (2004)

Chen, C.X., Zaniolo, C.: SQL^{ST}: A Spatio-temporal Data Model and Query Language. In: Proceedings of the 19th International Conference on Conceptual Modeling, pp. 96–111. Salt Lake City, Utah, USA (2000)

Chen, A., Di, L., Bai, Y., Wei, Y.: Grid computing and its application to geoinformatics. In: Karimi, H. (ed.) The Encyclopedia of Geoinformatics. Idea Group Inc., Hershey, PA, USA (2006, in press)

Chen, A., Di, L., Bai, Y., Wei, Y.: An optimized grid-based, OGC standards-compliant collaborative software system for serving NASA geospatial data. In: The 2nd IEEE Systems and Software Week (SASW 2006), 30th Annual Software Engineering Workshop. Loyola College Graduate Center, Columbia, MD, USA. April 25–27, 2006

Chen, A., Di, L., Wei, Y., Liu, Y., Bai, Y., Hu, C., Mehrotra, P.: Grid enabled geospatial catalogue web services. In: American Society for Photogrammetry and Remote Sensing 2005, Baltimore, MD, March 7–11, 2005

Chen, G., Kotz, D.: A survey of context-aware mobile computing research. Technical Report Dartmouth Computer Science Technical Report TR2000–381 (2000)

Cheng, M.Y., Ko, C.H., Chang, C.H.: Computer-aided DSS for safety monitoring of geotechnical construction. Aut. Const. **11**(4), 375–390 (2002)

Cheng, R., Kalashnikov, D., Prabhakar, S.: Evaluating probabilistic queries over imprecise data. In: Proc. ACM SIGMOD, 2003

Cheng, R., Kalashnikov, D.V., Prabhakar, S.: Querying imprecise data in moving object environments. IEEE TKDE, **16**(9), September 2004

Cheng, R., Singh, S., Prabhakar, S., Shah, R., Vitter, J., Xia, Y.: Efficient join processing over uncertain data. In: Proc. ACM CIKM, 2006

Cheng, R., Singh, S., Prabhakar, S.: U-DBMS: An uncertainty database management system. In: Proc. VLDB, 2004

Cheng, R., Xia, Y., Prabhakar, S., Shah, R., Vitter, J.S.: Efficient indexing methods for probabilistic threshold queries over uncertain data. In: Proc. VLDB, 2004

Cheng, R., Zhang, Y., Bertino, E., Prabhakar, S.: Preserving User Location Privacy in Mobile Data Management Infrastructures. In: Proceedings of Privacy Enhancing Technology Workshop. pp. 393–412. Cambridge, UK (2006)

Chen, H., Finin, T., Joshi, A. (2003) An Intelligent Broker for Context-Aware Systems. In: Ubicomp (2003), Seattle, Washington, USA. Available via. http://ebiquity.umbc.edu/paper/html/id/78/An-Intelligent-Broker-for-Context-Aware-Systems. Accessed 12–15 Oct 2003

Chen J., Cheng, R.: Efficient evaluation of imprecise location-dependent queries. In: Proc. IEEE ICDE, 2007

Chen, J., DeWitt, D.J., Tian, F., Wang, Y.: NiagaraCQ: A Scalable Continuous Query System for Internet Databases. ACM SIGMOD, Dallas, Texas, USA, 16–18 May 2000

Chen, J.K., Huang, Y.F., Chin, Y.H.: A Study of Concurrent Operations on R-Trees. Information Sciences: an International Journal. **98**(1–4), 263–300 (1997)

Chen, J., Larson, R.R., Stonebraker, M.: Sequoia 2000 object browser. In: Digest of Papers, COMPCON Spring 1992. The 37th IEEE Computer Society International Conference, Los Alamitos, US (1992)

Chen, L., Ng, R.T.: On the marriage of lp-norms and edit distance. In Proc. of VLDB, pp. 792–803 (2004)

Chen, L., Özsu, M.T., Oria, V.: Robust and fast similarity search for moving object trajectories. Proc. ACM SIGMOD Conf., Baltimore, Maryland, USA, pp. 491–502 (2005)

Chen, A.: MasterCard Cashes in on Web Services. http://www.eweek.com/article2/0,1759,1761448,00.asp (2005)

Chen, M., Chen, L., Miller-Hooks, E.: Traffic Signal Timing for Urban Evacuation. ASCE J. Urban Plann. Dev. – Special Emergency Transportation Issue **133**(1), 30–42

Chen, M., Chien, S.: Dynamic freeway travel time prediction using probe vehicle data: link-based vs path-based. J. Trans. Res. Board **1768**, 157–161 (2002)

Chen, P.S.: The Entity-Relationship Model: Toward a unified view of Data., ACM Trans. Database Sys. **1**, 9–36 (1976)

Chen, Y., Revesz, P.: Max-count aggregation estimation for moving points. 11th International Symposium on Temporal Representation and Reasoning,pp. 103–108. IEEE Press, Washington (2004)

Chen, Y., Yang, Q., Yin, J., Chai, X.: Power-efficient access-point selection for indoor location estimation. IEEE Trans. on Knowledge and Data Engineering **18**(7), 877–888 (2006)

Chern, H.-H., Fuchs, M., Hwang, H.-K.: Phase Changes in Random Point Quadtrees. National Taiwan Ocean University, National Chiao Tung University, and Institute of Statistical Science Academia Sinica, Taiwan. ACM Transact Algo 3(12):2 (2007)

Cherniack, M., Galvez, E.F., Franklin, M., Zdonik, S.: Profile-driven cache management. In: ICDE (2003)

Chervenak, A., Foster, I., Kesselman, C., Salisbury, C., Tuecke, S.: The data grid: towards an architecture for the distributed management and analysis of large scientific datasets. J. Network Comput. Appl. **23**, 187–200 (2001)

Chiang, Y., Tamassia R.: Dynamic Algorithms in Computational Geometry. Proc. of the IEEE, Special Issue on Computational Geometry **80**(9), 1412–1434 (1992)

Chica-Olmo, M., Abarca-Hernandez, F.: Computing geostatistical image texture for remotely sensed data classification. Comput. Geosci. (1999)

Chiles, J.P.: Delfiner, P.: Geostatistics: Modeling Spatial Uncertainty. John Wiley and Sons, New York (1999)

Chin, F.Y., Snoeyink, J., Wang, C.A.: Finding the medial axis of a simple polygon in linear time. In: ISAAC 95: Proceedings of the 6th International Symposium on Algorithms and Computation, pp. 382–391. Springer-Verlag, London, UK (1995)

Chiu, Y.C., Korada, P., Mirchandani, P.B.: Dynamic Traffic Management for Evacuation. Presented at The 84th Annual Meeting of Transportation Research Board, Washington, DC (2004)

Chiu, Y.C., Villlalobos, J.A., Mirchandani, P.B.: The Development of the Multi-Resolution Assignment and Loading of Traffic Activities (MALTA) for Regional Mass Evacuation Analysis in Central Texas, (2007)

Chiu, Y.C., Zhang, H., Villalobos, J.A., Gautam, B.: Modeling no-notice mass evacuation using a dynamic traffic flow optimization model. IIE Trans **39**, 83–94 (2007)

Chiu, Y.C., Zheng, H.: Real-Time Mobilization Decisions for Multi-Priority Emergency Response Resources and Evacuation Flows: Model Formulation and Solution. Transportation Research Part E (in press 2007)

Cho, H.J., Chung, C.W.: An Efficient and Scalable Approach to CNN Queries in a Road Network. VLDB (2005)

Choi, J.Y., et al.: Utilizing web-based GIS and SDSS for hydrological land use change impact assessment. Trans. ASAE **48**(2), 815–822 (2005)

Choi, J.Y., Jakobsson, M., Wetzel, S.: Balancing auditability and privacy in vehicular networks. In: Q2SWinet '05: Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks, Montreal, Quebec, Canada, 2005, pp. 79–87. ACM Press, New York, NY, USA (2005)

Choi, K.M., Serre, M.L., Christakos, G.: Efficient mapping of California mortality fields at different spatial scales. J. Expo. Anal. Environ. Epidemiol. **13**, 120–133 (2003)

Chomicki, J., Godfrey, P., Gryz, J., Liang, D.: Skyline with Presorting. In: proceedings of ICDE'03, IEEE Computer Society, pp. 717–816 Bangalore, India (2003)

Chomicki, J., Revesz, P.Z.: Constraint-Based Interoperability of Spatiotemporal Databases. In: Scholl, M., Voisard, A. (eds.) Advances in Spatial Databases, pp. 142–161. Springer-Verlag, Berlin, Heidelberg, New York (1997)

Chon, H.D., Agrawal, D., Abbadi, A.E.: Query Processing for Moving Objects with Space-Time Grid Storage Model. In: Proceedings of the 3rd Int. Conf. on Mobile Data Management, Singapore, 8–11 Jan 2002, pp. 121–126

Chow, C.Y., Mokbel, M.F., Liu, X.: A Peer-to-Peer Spatial Cloaking Algorithm for Anonymous Location-based Services. In: Proceedings of the ACM Symposium on Advances in Geographic Information Systems (ACM GIS), pp. 171–178. Arlington, Virginia, USA (2006)

Chrisman, N.: Charting the unknown: how computer mapping at Harvard became GIS. ESRI Press, Redlands, CA (2006)

Chrisman, N.R.: The Error Component in Spatial Data. In: Maguire, D.J., Goodchild, M.F., Rhind, D. W. (eds.) Geographical Information Systems, Longman, London (1991)

Christakos, G.: A Bayesian/maximum-entropy view to the spatial estimation problem. Math. Geol. **22**, 763–776 (1990)

Christakos, G., Bogaert, P., Serre, M.: Temporal GIS: Advanced Functions for Field-Based Applications. Springer, New York (2002)

Christakos, G., Kolovos, A., Serre M.L., Vukovich, F.: Total ozone mapping by integrating data bases from remote sensing instruments and empirical models. IEEE Trans. Geosci. Remote Sensing **42**, 991–1008 (2004)

Christakos, G., Li, X.: Bayesian maximum entropy analysis and mapping: A farewell to kriging estimators? Math. Geol. **30**, 435–462 (1998)

Christakos, G.: Modern Spatiotemporal Geostatistics. Oxford University Press, New York (2000)

Christakos, G., Olea, R.A., Serre, M.L., Yu, H.L., Wang, L.-L.: Interdisciplinary Public Health Reasoning and Epidemic Modelling: The Case of Black Death. Springer-Verlag, New York (2005)

Christakos, G.: On certain classes of spatiotemporal random fields with application to space-time data processing. IEEE Trans. Syst. Manuf. Cybernetics **21**, 861–875, (1991)

Christakos, G.: On a deductive logic-based spatiotemporal random field theory. Probability Theor. Math. Stat. (Teoriya Imovirnostey ta Matematychna Statystyka) **66**, 54–65 (2002)

Christakos, G.: Random field modelling and its applications in stochastic data processing. Ph.D. Thesis, Harvard University (1990)

Christakos, G.: Random Field Models in Earth Sciences. Academic, San Diego (1992)

Christian, E. (2004) The architecture of GEOSS. http://usgeo.gov/docs/GEOSSArchitecture.ppt. Accessed 25 May 2007

Christie, M., Subbey, S., Sambridge, M.: Prediction under uncertainty in reservoir modeling. In: the European Conference on the Mathematics of Oil Recovery, (2002)

Christine, P., Spaccapietra, S., Zimnyi, E.: Spatio-temporal conceptual models: data structures + space + time. In: GIS '99: Proceedings of the 7th ACM international symposium on Advances in geographic information systems, pp. 26–33, ACM Press, New York, NY, USA (1999)

Christofides, N., Mingozzi, A., Toth, P.: The vehicle routing problem. In: Christofides, N., Mingozzi, A., Toth, P., Sandi, C. (eds.) Combinatorial Optimization, pp. 315–338. Wiley, Chichester (1979)

Christofides, N.: The optimum traversal of a graph, Int. J. Manage. Sci. **1**(6) (1973)

Christofides, N.: Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Computer Science Department, CMU (1976)

Christ, T.W., Godwin, P.A., Lavigne, R.E.: A prison guard duress alarm location system. In: Proceedings of the IEEE International Carnahan Conference on Security Technology, vol.1, pp. 106–116, Ottawa, Ont., Canada (1993)

Chung, C., Chun, S., Lee, J., Lee, S.: Dynamic Update Cube for Range-Sum Queries. In: VLDB, pp. 521–530 (2001)

Chun, S.A., Atluri, V.: Protecting privacy from continuous high-resolution satellite surveillance. In: Proceedings of the 14th IFIP 11.3 Working Conference on Database Security (IFIP WG11.3 2000), Scoorl, The Netherlands, August 21–23, 2000, pp. 399–420 (2000)

Ciaccia, P., Patella, M., Zezula, P.: M-tree: An efficient access method for similarity search in metric spaces. In: VLDB '97: Proceedings of the 23rd International Conference on Very Large Data Bases, pp. 426–435. Morgan Kaufmann Publishers Inc, San Francisco, CA (1997)

Ciolli,M. Zatelli, P.: http://www.ing.unitn.it/~grass/conferences/grass2002/proceedings/proceedings/abstracts.html. online, 12 (2006)

Civilis, A., Pakalnis, S.: Techniques for efficient road-network-based tracking of moving objects. IEEE Trans. Knowl. Data Eng. **17**(5), 698–712 (2005) Senior Member-Christian S. Jensen

Clarke, R.V.: Situational Crime Prevention: Successful Case Studies. Harrow and Heston, New York (1992)

Clark, I.: Practical Geostatistics. Applied Science, (1979)

Clark, P.J., Evans, F.C.: Distance to nearest neighbor as a measure of spatial relationships in populations. Ecology **35**, 445–453 (1954)

Clementini, E., Di Felice, P.: An algebraic model for spatial objects with indeterminate boundaries. In: Burrough, P.A., Frank, A.U. (eds.) Geographic Objects with Indeterminate Boundaries, GISDATA Series, pp. 155–169. Taylor & Francis, London (1996)

Clementini, E., Di Felice, P.: Approximate topological relations. Int J Approx Reason **16**, 173–204 (1997)

Clementini, E., Di Felice, P., Califano, G.: Composite Regions in Topological Queries. Information Systems **20**, 579–594 (1995)

Clementini, E.,Di Felice, P.A.: Comparison of Methods for Representing Topological Relationships. Information Sciences **80**, 1–34 (1994)

Clementini, E., Di Felice, P.: A model for representing topological relationships between complex geometric features in spatial databases. Inf Sci **90**, 121–136 (1996)

Clementini, E., Di Felice, P.: A spatial model for complex objects with a broad boundary supporting queries on uncertain data. Data Knowl. Eng. **37**(3), 285–305, (2001)

Clementini, E., Di Felice, P., van Oosterom, P.: A small set of formal topological relationships suitable for end-user interaction. In: Abel, D., Ooi, B.C. (eds.) Advances in Spatial Databases – Third International Symposium, SSD '93. Lecture Notes in Computer Science, vol. 692, pp. 277–295. Springer, Berlin (1993)

Clementini, E.: A model for lines with a broad boundary. In: 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2002),Annecy, France, number 692 in LNCS, pp. 1579–1586 (2002)

Clementini, E.: A model for uncertain lines. J Vis Language Comput **16**, 271–288 (2005)

Clerici, A., Perego, S., Tellini, C., Vescovi, P.: A procedure for landslide susceptibility zonation by the conditional analysis method. Geomorphol. **48**(4), 349–364 (2002)

Cliff, A.D., Ord, J.K.: Spatial Autocorrelation. Pion, International Co-Operative Publishing House, London, UK (1973)

Cliff, A.D., Ord, J.K.: Spatial Autocorrelation. Pion, London (1973)

Cliff, A., Ord, J.K.: Spatial Processes, Models and Applications. Pion, London (1981)

Clough, P.: Extracting metadata for spatially-aware information retrieval on the internet. In: Proceedings of the 2005 Workshop on Geographic Information Retrieval, Bremen, Germany 2005

Coad, P., Yourdon, E.: Object-Oriented Analysis. 2nd edn. Prentice-Hall, New York (1991)

Cobb, M., Chung, M.J., Foley III, H., Petry, F.E., Shaw, K.B.: A Rule-based Approach for the Conflation of Attributed Vector Data. Geoinformatica **2**(1), 7–35 (1998)

Codd, E.F.: Twelve rules for on-line analytic processing. Comput. (1995)

Cohen J., Small, C.: Hypsographic demography: the distribution of human population by altitude. Proc. Natl. Acad. Sci. **95**, 14009–14014 (1998)

Cohen, L.E., Felson, M.: Social change and crime rate trends: a routine activity approach. Am. Soc. Rev. **44**, 588–608 (1979)

Cohn, A.G., Gotts, N.M.: The 'egg-yolk' representation of regions with indeterminate boundaries. In: Burrough, P., Frank, A.M. (eds.): Proceedings, GISDATA Specialist Meeting on Geographical Objects with Undetermined Boundaries, pp. 171–187. Taylor Francis, London (1996)

Cohn, A.G., Gotts, N.M.: The 'Egg-Yolk' Representation of Regions with Indeterminate Boundaries. In: Geographic Objects with Indeterminate Boundaries, pp. 171–187. Taylor & Francis, London, UK (1996)

Cohn, A.G., Gotts, N.M.: A mereological approach to representing spatial vagueness. In: Doyle, J. Aiello, L.C., Shapiro, S. (eds.): Principles of Knowledge Representation and Reasoning, Proc. 5th Conference, pp. 230–241. Morgan Kaufmann, San Mateo (1996)

Cohn, A.G., Hazarika, S.M.: Qualitative Spatial Representation and Reasoning: An Overview. Fundam. Inf. **46**(1–2), 1–29 (2001)

Cohn, A.G., Renz, J.: Qualitative Spatial Representation and Reasoning. In: Lifschitz, V., van Harmelen, F., Porter, F. (eds.) Handbook of Knowledge Representation, Ch. 13. Elsevier, München (2007)

Coleman, K.: Convergence: GIS/Communications/Information Technologies. In: *Direction Mag.* **13/11/2004**. Available via: www.directionsmag.com (2004)

Collinge, S.K., Ray, C. (eds.): Disease Ecology: Community Structure and Pathogen Dynamics. Oxford University Press, Oxford (2006)

Comber, A., Fisher, P., Wadsworth, R.: Integrating land cover data with different ontologies: identifying change from inconsistency. Int. J. Geogr. Inf. Sci. **18**, 691–708 (2004)

Comer, D.: The ubiquitous B-tree. ACM Computing Surveys **11**(2):121–137 (1979)

Commercial Exploitation of Europe's Public Sector Information. PIRA International, Leatherhead, UK (2000)

Cooke, D., Maxfield, W.: The development of a geographic base file and its uses for mapping. In: 5th Annual Conference of the Urban and Regional Information System Association (URISA), pp. 207–218 (1967)

Coppin, P., Jonckheere, I., Nackaerts, K., Muys, B., Lambin, E.: Digital change detection methods in ecosystem monitoring: a review. Int. J. Remote Sens. **25**(9), 1565–1596 (2004)

Cormack, R.M., Ord, J.K.: Spatial and Temporal Analysis in Ecology. International Co-Operative Publishing House, Fairfield, USA (1979)

Cormen, T.H., Leiserson, C.E., Rivest, R.: Introduction to Algorithms. McGraw-Hill, New York (1999)

Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to Algorithms. MIT Press, McGraw-Hill, Cambridge, London (1990)

Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. MIT Press. Cambridge, MA (2001)

Cormen, T., Leiserson, C., Rivest, R., Stein, C.: Introduction to Algorithms. The MIT Press (1997)

Cornell Law School Legal Information Institute: http://www.law.cornell.edu/uscode/html/uscode10/usc_sec_10_00000467----000-.html (2006)

Cornford, D., Csató, L., Opper, M.: Sequential, Bayesian geostatistics: a principled method for large data sets. Geogr. Anal. **37**, 183–199 (2005)

Cornish, D., Clarke, R.V.: The Reasoning Criminal. Springer-Verlag (1986)

Cornwell, J., et al.: Affordance Theory for Improving the Rapid Generation, Composability, and Reusability of Synthetic Agents and Objects. 12th Conference on Behavior Representation in Modeling and Simulation (2003)

Coroama, V., Bohn, J., Mattern, F.: Living in a Smart Environment – Implications for the Coming Ubiquitous Information Society, In: Proceedings of the International Conference on Systems, Man and Cybernetics 2004 (IEEE SMC 2004) 5633–5638, http://citeseer.ist.psu.edu/coroama04living.html

Corral, A., Manolopoulos, Y., Theodoridis, Y., Vassilakopoulos, M.: Closest Pair Queries in Spatial Databases. ACM SIGMOD, Dallas, Texas, USA, 16–18 May 2000

Costa, J.A., Patwari, N., Hero III, A.O.: Distributed multidimensional scaling with adaptive weighting for node localization in sensor networks. IEEE/ACM Trans. Sens. Net. **2**(1), 39–64 (2006)

Cotelo Lema, J.A., Forlizzi, L., Güting, R.H., Nardelli, E., Schneider, M.: Algorithms for moving object databases. Comp. J. **46**(6), 680–712 (2003)

Cotter, C.H.: The Astronomical and Mathematical Foundations of Geography. American Elsevier, New York (1966)

Couclelis, H.: People manipulate objects (but cultivate fields): beyond the raster-vector debate in GIS. In: Theories and Methods of Spatial-Temporal Reasoning in Geographic Space. Lecture Notes in Computer Science, vol. 639, pp. 65–77. Springer, Berlin (1992)

Cover, T.M., Hart, K.J.: Nearest Neighbor Pattern Classification. IEEE Transaction on Inf. Theory. **13**(1), 21–27 (Jan, 1967)

Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley, Chichester, U.K. (1991)

Covington, H.J., Fogla, P., Zhan, Z., Ahamad, M.: A context-aware security architecture for emerging applications. In: Proceedings of the18th Annual Computer Security Applications Conference (ACSAC '02), December 10–14, 2007, Miami Beach, Florida 2002.

Covington, M. J., Long, W., Srinivasan, S., Dev, A. K., Ahamad, M., Abowd, G. D.: Securing context-aware applications using environment roles. In: Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies. SACMAT '01, May 3–4, 2001, Litton-TASC, Chantilly, Virginia, USA 2001

Cox, J., Kautz, R., MacLaughlin, M., Gilbert, T.: Closing the gaps in Florida's wildlife habitat conservations systems. Office of Environmental Services, Tallahassee, Florida (1994)

Cox, S., Daisey, P., Lake, R., Portele, C., Whiteside, A.: OpenGIS Geography Markup Language (GML 3.0) Implementation Specification, OpenGIS Specifications. http://www.opengis.org/specs/?page=specs. Accessed 3 Dec 2006

Craglia, M., Masser, I.: Access to Geographic Information: a European Perspective. J. Urban Regional Inf. Syst. Assoc. (URISA) **15**(APA I), 51–59 (2003)

Cramer, M.: EuroSDR network on digital camera calibration. IAPRS, Vol. XXXV, Part B, Istanbul, digitally on CD-Rom, (2004) 6 pages

Cramer, M.: Performance of medium format aerial sensor systems, IAPRS, vol. XXXV, part B, Istanbul, digitally on CD-Rom (2004) 6 pages

Cremer, F., den Breejen, E., Schutte, K.: Sensor Data Fusion for Antipersonnel Land Mine Detection. In: Proceedings of EuroFusion98, pp. 55–60. Great Malvern, UK (1998)

Cressie, N.A.C.: Statistics for Spatial Data. Wiley & Sons, New York (1993)

Cressie, N., Huang, H.-C.: Classes of nonseparable spatio-temporal stationary covariance functions. J. Am. Stat. Assoc. **94**, 1330–1340 (1999)

Cressie, N., Read, T.R.C.: Do Sudden Infant Deaths Come in Clusters?. Statistics and Decisions, Supplement Issue **2**(3), 333–349 (1985)

Cressie, N.: Statistics for Spatial Data. Wiley and Sons, Inc., New York (1991)

Cresswell, T.: Place: A Short Introduction. Blackwell Publishing Ltd (2004)

Cromley, E.K., McLafferty, S.L.: GIS and Public Health. Guilford Publications Inc, New York (2002)

Cron, J.: Graphische Benutzeroberflächen interaktiver Atlanten – Konzept zur Strukturierung und praktischen Umsetzung der Funktionalität; mit Kriterienkatalog. Diploma thesis, Hochschule für Technik und Wirtschaft, Dresden and Institute o Cartography, ETH Zurich (2006). Available via http://www.ika.ethz.ch/teaching/Diplomarbeit-Cron.pdf; http://www.ika.ethz.ch/teaching/Diplomarbeit-Cron-Katalog.pdf

Crosetto, M., Tarantola, S.: Uncertainty and sensitivity analysis: tools for GIS-based model implementation. Int. J. Geogr. Inf. Sci. **15**, 415–437 (2001)

Crossland, M.D.: Individual decision-maker performance with and without a geographic information system: an empirical study. Unpublished doctoral dissertation, Kelley School of Business. Indiana University, Bloomington, IN (1992)

Crossland, M.D., Wynne, B.E., Perkins, W.C.: Spatial decision support systems: an overview of technology and a test of efficacy. Decis. Supp. Sys. **14**(3), 219–235 (1995)

CubeWerx Inc. http://www.cubewrex.com/. Accessed 20 Aug 2006

Cuellar, J., Morris, J., Mulligan, D., Peterson, J., Polk, J.: Geopriv requirements. RFC 3693, IETF (2004)

Cui, N., Hong, L., Layne, J.R. A comparison of nonlinear filtering approaches with an application to ground target tracking. Signal Process. **85**, 1469–1492 (2005)

Cullmann, J., Mishra, V., Peters, R.: Flow analysis with wasim-eth model parameter sensitivity at different scales. Adv. Geosci. **9**, 73–77 (2006)

Culshaw, M.G., et al.: The role of web-based environmental information in urban planning – the environmental information system for planners. Sci. Total Env. **360**(1–3), 233–245 (2006)

Cycorp OpenCyc. http://www.cyc.com/opencyc. Accessed 21 March 2007

Dagan, S.: Math Animated – A hosted teaching aid using graphic animations and interactivity (2005), http://www.mathanimated.com/, accessed 2006–11–12

Daley, R.: Atmospheric Data Analysis. Cambridge University Press (1996)

Damasio, A.: The Feeling of What Happens. First Harvest edition, San Diego, New York, London (1999)

Dangermond, J.: Speaking the Language of Geography – GIS, ArcNews. Fall (2004). http://www.esri.com/news/arcnews/fall04articles/speaking-the-language1of2.html

Danko, D.: Senior Consultant, GIS Standards, Environmental Systems Research Institute, ISO Metadata Workshop

Dare, P., Dowman, I.: A new approach to automatic feature based registration of SAR and SPOT images. Int. Arch. Photogramm. Remote Sens. **33**(B2), 125–130 (2000)

Darken, R.P., Peterson, B.: Spatial Orientation, Wayfinding, and Representation. In: Stanney, K.M. (ed.) Handbook of Virtual Environments: Design, Implementation, and Applications, pp. 493–518. Lawrence Erlbaum Associates, Mahwah NJ (2002)

Das, G., Gunopulos, D., Mannila, H.: Finding Similar Time Series. In Proc. of PKDD, pp. 88–100 (1997)

Das, G., Lin, K., Mannila, H., Renganathan, G., Smyth, P.: Rule discovery from time series. In: Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, New York, 27–31 August 1998, pp. 16–22

Da Silva Ramos, C., Cartwright, W.,: Atlases from paper to digital medium. In: Stefanakis, E., Peterson, M.P., Armenakis, C., Delis, V.(eds.) Geographic Hypermedia–Concepts and Systems. Lecture Notes in Geoinformation and Cartography:, pp. 97–19. Springer, Berlin Heidelberg New York (2006)

Datar, M., Immorlica, N., Indyk, P., Mirrokni, V.: Locality sensitive hashing scheme based on p-stable distributions. In: Proceedings of the ACM Symposium on Computational Geometry, pp. 253–262 (2004)

Datta, A., Quarteroni, S., Aberer, K.: Autonomous Gossiping: A self-Organizing epidemic algorithm for selective information dissemination in wireless mobile ad hoc networks. In: The International Conference on Semantics of a Networked World (2004)

Daubechies, I.: Ten Lectures on Wavelets. Soc. Indust. Appl. Math., Philadelphia (1992)

David, B., Raynal, L., Schorter, G.: GeO2: Why objects in a geographical DBMS? In: Abel, D.J., Ooi, B.C. (eds.) Proceedings of the 3rd International Symposium on Advances in Spatial Databases, SSD'93, Singaporer, 23–25 June 1993. Lecture Notes in Computer Science, vol. 692, pp. 264–276. Springer Berlin, Heidelberg (1993)

David, E., Erickson, J.: Raising roofs, crashing cycles, and playing pool: applications of a data structure for finding pairwise interactions. In: SCG 98: Proceedings of the 14th Annual Symposium on Computational Geometry, 58–67 (1998)

Davies, J., Fensel, D., van Harmelen, F.: Towards the Semantic Web: Ontology-Driven Knowledge Management. Wiley, UK (2003)

Davies, N., Gellersen, H.W.: Beyond prototypes: challenges in deploying ubiquitous systems, Pervasive Computing, IEEE **1**(1), 26–35 (2002)

Davis, C.A., Fonseca, F.: Considerations from the development of a local spatial data infrastructure. Inf. Technol. Dev. **12**, 273–290 (2006)

Davis, F. W.: Gap analysis for the southwestern California region. Technical Report 94–4. National Center for Geographic Information and Analysis, Santa Barbara, California (1994)

Davis, M., Aquino, J.: JTS Topology Suite – Technical Specifications. Vivid Solutions Victoria, British Columbia (2003)

DeBarry, P.A., Garbrecht, J., Garcia, L., Johnson, L.E., Jorgeson, J., Krysanova, V., Leavesley, G., Maidment, D., Nelson, E.J., Ogden, F.L., Olivera, F., Quimpo, R.G., Seybert, T.A., Sloan, W.T., Burrows, D., Engman, E.T.: GIS Modules and Distributed Models of the Watershed, p. 120. American Society of Civil Engineers Water Resources Division – Surface Water Hydrology Committee, Reston, VA (1999)

DeBarry, P.A.: Watersheds: processes, assessment, and management. John Wiley and Sons, Inc., Hobokon NJ (2004)

Debenham, J., Clarke, G., Stillwell, J.: Extending geodemographic classification: a new regional prototype. Environ. Plann. A **35**(6), 1025–1050 (2003)

Debregeas, A., Hebrail, G.: Interactive interpretation of kohonen maps applied to curves. In: Proceedings of the 4th International Conference of Knowledge Discovery and Data Mining, New York, 27–31 August 1998, pp. 179–183

Dechter, R., Pearl, J.: Generalized best-first search strategies and the optimality of A*. J. ACM **32**(3), 505–536 (1985)

Defence Intelligence Collection Group (ICG) in the United Kingdom. http://www.mod.uk/DefenceInternet/AboutDefence/WhatWeDo/SecurityandIntelligence/DIS/ICG/

Defense Imagery and Geospatial Organization (DIGO) in Australia. http://www.defence.gov.au/digo/

Deichmann, U., Balk, D., Yetman, G. Transforming Population Data for Interdisciplinary Usages: from Census to Grid. http://sedac.ciesin.columbia.edu/gpw-v2/GPWdocumentation.pdf (2001). Accessed 17 Sept 2007

Delis, V., Hadzilacos, T.H., Tryfona, N.: An Introduction to Layer Algebra. In: Proceeding of the 6th International Conference on Spatial Data Handling (SDH). Edinburgh, Scotland 5"9 Sept 1995

Demcsak MF (1997) HDF-EOS Library Users Guide for the ECS Project Volume 2: Function Reference Guide. Hughes Applied Information Systems. doi:170-TP-006-003

Demers, A.J., Greene, D.H., Hauser, C., Irish, W., Larson, J., Shenker, S., Sturgis, H.E., Swinehart, D.C., Terry, D.B.: Epidemic Algorithms for Replicated Database Maintenance. PODC 1–12 (1987)

DeMers, M.: Fundamentals of geographic information systems, 3rd ed. John Wiley & Sons, New York (2005)

DeMers, M.N.: GIS Modeling in Raster. Wiley, New York (2002)

Demirbas, M., Ferhatosmanoglu, H.: Peer-to-peer spatial queries in Sensor Networks. In: Paper presented at the 3rd IEEE International Conference on Peer-to-Peer Computing (P2P'03), pp. 32–39. Linkoping, Sweden, September 2003

Dempsey, M.: The fundamentals of indoor positioning systems in healthcare. Biomed. Instrum. Technol. 293–297 (2003), (PDF version on the web: www.radianse.com/download/news-bit-2004.pdf, July 12th, 2007)

Denning, D.E., MacDoran, P.F.: Location-based authentication: grounding cyberspace for better security. Computer Fraud and Security Bulletin, February 1996, pp. 12–16 (1996)

Densham, P.J.: Spatial decision support systems. In: Maguitre, D.J., Goodchild, M.F., Rhind, D. (eds.) Geographical information systems: Principles and applications, vol. 1, pp. 403–412, Longman, London (1991)

Dent, B.D.: Cartography, 4th edn. Wm. C. Brown, Dubuque (1996)

Deuchler, C., Wählisch, M., Gehrke, S., Hauber, E., Oberst, J., Jaumann, R.: Combining mars data in grass GIS for geological mapping. ISPRS Congress, 12–23 July 2004 Istanbul, Turkey, vol.#160;35, ISPRS, Istanbul (2004)

Devary, J.L., Rice, W.A.: Geostatistis software user's manual for geosciences research and engineering. Pacific Northwest Laboratory, Richland, Washington (1982)

DeVogele, T., Parent, C., Spaccapietra, S.: On spatial database integration. Int. J. Geogr. Inf. Sci. **12**, 335–352 (1998)

DeVogele, T., Trevisan, J., Ranal, L.: Building a Multi Scale Database with Scale Transition Relationships. In: Kraak, M.-J., Molenaar, M., Fendel, E.M. (eds.) Advances in GIS Research II, Proceedings of the 7th International Symposium on Spatial Data Handling, pp. 337–352. Taylor and Francis, Delft, London, (1996)

Dey, A.K.: Understanding and using context. Personal Ubiquitous Comput. J. **5**, 4–7 (2001)

Dhamankar, R., Lee, Y., Doan, A., Halevy, A.Y., Domingos, P.: iMAP: discovering complex mappings between database schemas. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), 2004, June 13–18, Paris, France

Dhillon, I., Fan, J., Guan, Y.: Efficient Clustering of Very Large Document Collections. In: Grossman, R., Kamath, C., Kegelmeyer, P., Kumar, V., Namburu, R. (eds.) Data Mining for Scientific and Engineering Applications. Kluwer Academic Publishers, Dordrecht, The Netherlands (2001)

Dias, E., Beinat, E., Scholten, H.: Effects of Mobile Information Sharing in Natural Parks. Paper presented at ENVIROINFO 2004, 18th International Conference Informatics for Environmental Protection, CERN, Geneva, Switzerland, 2004

Diaz, J., Penrose, M., Petit, J., Serna, M.: Convergence theorems for some layout measures on random lattice and random geometric graphs. Combinatorics Probability Comput. **9**, 489–511 (2000)

Dibiase, D., Maceachren, A.M., Krygier, J.B., Reeves, C.: Animation and the role of map design in scientific visualization. Cartogr. Geogr. Inf. Syst. **19**, 201–214 (1992)

Dietzel, C., Clarke, K.C.: Decreasing computational time of urban cellular automata through model portability. GeoInformatica **10**, 197–211 (2006)

Diggle, P.J., Chetwynd, A., Haggvist, R., Morris, S.: Second-order analysis of space-time clustering. Stat. Methods Med. Res. **4**, 124–136 (1995)

Diggle, P.J.: Point process modellng in environmental epidemiology. In: Barnett, V., Turkman, K.F. (eds.) Statistics in the Environment SPRUCE I. Wiley, New York (1993)

Diggle, P.J., Ribeiro Jr, P.J.: Bayesian Inference in Gaussian Model-based Geostatistics. Geogr. Environ. Modelling **6**, 29–146 (2002)

Diggle, P.J., Ribeiro, P.J.: Model-based Geostatistics. Springer, New York (2007)

Diggle, P.J.: Statistical Analysis of Point Processes. Chapman and Hall, New York (1983)

Diggle, P.J.: Statistical Analysis of Spatial Point Patterns. Mathematics in Biology. Academic, London (1983)

Diggle, P.J., Tawn, J.A., Moyeed, R.A.: Model-based geostatistics (with discussions). Appl. Stat. **47**(3), 299–350 (1998)

Digital Geographic Information Exchange Standard (DIGEST), Version 1.2, https://www.dgiwg.org/digest, Digital Geographic Information Working Group, January (1994)

Dijkstra, E.W.: A Note on Two Problems in Connection with Graphs. Numeriche Mathematik, vol. 1, 269–271 (1959)

Dikau R., Saurer, H.: GIS for Earth Surface Systems. Gebruder Borntraeger (1999)

Di, L., Chen, A., Bai, Y., Wei, Y.: Implementation of geospatial product virtualization in grid environment. In: The Sixth Annual Earth Science Technology Conference – ESTC2006, College Park, MD, June 27–29, 2006

Di, L., Chen, A., Yang, W., Liu, Y., Wei, Y., Mehrotra, P., Hu, C., Williams, D.: The development of a geospatial data grid by integrating OGC Web Services with Globus-based grid technology. Concurrency and Computation: Practice and Experience (2006)

Di, L.: The development of remote sensing related standards at FGDC, OGC, and ISO TC211. In: Proceedings of IEEE International Geoscience and Remote Sensing Symposium, Toulouse, France, 21–25 July 2003

Di, L.: Geospatial grids, geospatial modeling, and virtual geospatial products. In: Earth Observation and Geospatial Technology for Civil Society, University College London, London, 23–25 June 2004

Di, L.: The geospatial grid. In: S. Rana, J. Sharma (eds.) Frontiers of Geographic Information Technology, pp 121–138. Springer-Verlag, Berlin, Heidelberg (2006)

Di, L., Kresse, W., Kobler, B.: The New ISO TC211 Standard Project on Sensor and Data Model for Imagery and Gridded Data. In: Proceedings of International Geoscience and Remote Sensing Symposium (IGARSS 2001). Sdyney, Australia, 2001

Di, L., Kresse, W., Kobler, B.: The Progress and Current Status of ISO Standard Development Project 19130-Sensor and Data Models for Imagery and Gridded Data. In: Chen, J., Jiang, J. (eds.) The International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences, Vol. 34, Part 2, Commission II, pp. 107-110. International Society for Photogrammetry and Remote Sensing (2002)

Di, L., Suresh, R., Ilg, D., Meyer, T.: EOSDIS Data Models and Example of Implementations. In: Proceedings of The Third International Conference/Workshop on Integrating GIS and Environmental Modeling (CD-ROM and WWW), Santa Fe, New Mexico, 21–25 Jan 1996

Di Luzio, M., Srinivasan, R., Arnold, J.G.: A GIS-coupled hydrological model system for the watershed assessment of agricultural nonpoint and point sources of pollution. Transactions in GIS **8**(1), 113–136 (2004)

Di, L., Yang, W., Chen, A., Liu, Y., Wei, Y., Mehrotra, P., Hu, C., Williams, D.: The development of a geospatial grid by integrating OGC Web Services with Globus-based grid technology. In: GridWorld/GGF15, Boston, MA, October 3–6, 2005

Di, L., Zhao, P., Yang, W., Yu, G., Yue, P.: Intelligent Geospatial Web Services. In: Proceedings of IEEE 25th Anniversary International Geoscience and Remote Sensing Symposium. Seoul, Korea (2005)

Ding, H., Trajcevski, G., Scheuermann, P.: Efficient maintenance of continuous queries for trajectories. GeoInformatica (2007). doi: 10.1007/s10707-007-0029-9

Ding, Z. and Güting, R.H.: Managing moving objects on dynamic transportation networks. In: International Conference on Scientific and Statistical Database Management (SSDBM) (2004)

Ding, Z. and Güting, R.H.: Uncertainty management for network constrained moving objects. In: DEXA (2004)

Diniz, J.A.F., Bini, L.M., Hawkins, B.A.: Spatial auto-correlation and red herrings in geographical ecology. Glob. Ecol. Biogeogr. **12**, 53–64 (2003)

Di Stefano, G., Graziosi, F., Santucci, F.: Distributed Positioning Algorithm for Ad-Hoc Networks. In: Proceedings of the IEEE International Workshop on UWB Systems. Oulu, Finnland, June 2003

Diversitas – Integrating biodiversity science for human well-being. www.diversitas-international.org/. Accessed November 2006

Diversity in America, Supplement to American Demographics, Alison Stein Wellner, November, (2002) http://resource.mapinfo.com/static/files/document/1074653455507/diversity.pdf and www.demographics.com

Doan, A., Domingos, P., Halevy, A.: Reconciling schemas of disparate data sources: a machine learning approach. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD) 2001, May 21–24. Santa Barbara, CA

Doan, A.,, Halevy, A.Y.: Semantic Integration Research in the Database Community: A Brief Survey. AI Magazine **26**(1), 83–94 (2005)

Doan, A., Madhavan, J., Domingos, P., Halvey, A.: Learning to map between ontologies on the semantic web. In: Proceedings of WWW'02, Hanau, 7–11 May 2002

Dobson, J., Bright, E., Coleman, P., Bhaduri, B.: LandScan2000: A New Global Population Geography, Remotely-Sensed Cities. Taylor and Francis, London, UK (2003)

Dobson, J., Bright, E., Coleman, P., Durfee, R., Worley, B.: LandScan: a global population database for estimating populations at risk. Photogramm. Eng. Remote Sensing **66**, 849–857 (2000)

Dobson, J. E.: Automated geography. The Professional Geogr. **35**(2), 135–143 (1983)

Dobson, J.E., Fisher, P.F.: Geoslavery. IEEE Technology and Society Magazine **22**(1), 47–52 (2003)

DOJO: DOJO – the javascript toolkit (2005), http://dojotoolkit.org/, accessed 2006–11–12

Dolan, M.E., Holden, C.C., Beard, M.K., Bult, C.J.: Genomes as geography: using GIS technology to build interactive genome feature maps. BMC Bioinf. **7**, 416 (2006)

Döllner, J., Buchholz, H., Nienhaus, M., Kirsch, F.: Illustrative visualization of 3D city models. In: Proceedings of Visualization and Data Analysis 2005 (Electronic Imaging 2005, SPIE Proceedings), pp. 42–51 (2005)

Döllner, J., Walther, M.: Real-time expressive rendering of city models. In: Seventh International Conference on Information Visualization, pp. 245–251, London (2003)

Dolog, P., Henze, N., Nejdl, W., Sintek, M.: (2003) Towards the adaptive semantic web. In: Proceedings of the 1st Workshop on Principles and Practice of Semantic Web Reasoning (PPSWR'03), co-located with International Conference on Logic Programming (ICLP'03), Mumbai, India, December 8, 2003

Dolog, P., Henze, N., Nejdl, W., Sintek, M.: Personalization in Distributed e-Learning Environments. In: Proceedings of 13th World Wide Web Conference, New York, USA 17–22 May 2004

Domencich, T., McFadden, D.L.: Urban Travel Demand: A Behavioral Analysis. North-Holland Publishing Co., 1975. Reprinted 1996. Available at: http://emlab.berkeley.edu/users/mcfadden/travel.html (1975)

Domingos, P., Hulten, G.: Mining high-speed data streams. In: Proceedings of ACM Special Interest Group on Knowledge Discovery and Data Mining, Boston, Massachusetts, 2000, pp. 71–80

Dorling, D., Fairbairn D.: Mapping: ways of representing the world. Longman, London (1997)

Douaik, A., van Meirvenne, M., Toth, T., Serre, M.L.: Space-time mapping of soil salinity using probabilistic BME. J Stochastic Environ. Res. Risk Assess. **18**, 219–227 (2004)

Doucet, A., de Freitas, N., Gordon, N. (eds.): Sequential Monte Carlo in Practice. Springer-Verlag, New York, NY (2001)

Douglas, D.H., Peucker, T.K.: Algorithms for the reduction of the number of points required to represent a digitised line or its caricature. Can. Cartogr. **10**(2), 112–122 (1973)

Dragicevic, S., Balram, S.: A Web GIS collaborative framework to structure and manage distributed planning processes. J. Geogr. Syst. **6**, 133–153 (2004)

Driscoll, J.R., Sarnak, N., Sleator, D.D., Tarjan, R.E.: Making data structures persistent. J. Comput. Syst. Sci. **38**(1), 86–124 (1989)

Driscoll, J., Sarnak, N., Sleator, D., Tarjan, R.E.: Making data structures persistent. In: STOC (1986)

Dubin, R.A.: Predicting House Prices Using Multiple Listings Data. Journal of Real Estate Finance and Economics **17**(1), 35–60 (1998)

DublinCore Dublin Core Metadata Initiative. http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide1-0.html. Accessed 21 March 2007

Dublin Core. http://en.wikipedia.org/wiki/Dublin_Core. Accessed 13 Sept 2007

Ducke, B.: Archaeological predictive modelling in intelligent network structures. Doerr, M., Sarris, A. (eds.): CAA 2002 The Digital Heritage of Archaeology. Proceedings of the 30th CAA conference held at Heraklion, Crete, Greece, 2–6 April 2002, pages 267–273, Athens (2003)

Duckham, M., Nittel, S., Worboys, M.: Monitoring dynamic spatial fields using responsive geosensor networks. In: GIS '05: Proceedings of the 13th Annual ACM International Workshop on Geographic Information Systems, Bremen, Germany, 4–5 Nov 2005

Duckham, M., Nittel, S., Worboys, M.: Monitoring dynamic spatial fields using responsive geosensor networks. Shahabi, C., Boucelma, O. (eds.) ACM GIS 2005, pp. 51–60. ACM Press, New York (2005)

Duckham, M.: A user-oriented perspective of error-sensitive GIS development. Trans. GIS **6**(2), 179–194 (2002)

Duczmal L., Renato, A.: A simulated annealing strategy for the detection of arbitrarily shaped spatial clusters. Comput. Statist. Data Anal. **45**(2), (2004)

Dueck, G.: New optimization heuristics: the great deluge algorithm and the record to record travel. J. Comput. Phys. **104**, 86–92 (1993)

Dueck, G., Scheuer, T.: Threshold accepting. A general purpose optimization algorithm appearing superior to simulated annealing. J. Comput. Phys. **90**, 161–175 (1990)

Dueker, K., Butler, J.A.: GIS-T Enterprise Data Model with Suggested Implementation Choices, J. Urban Reg. Inform. Syst. Assoc. **10**(1), 12–36 (1998)

Duentsch, I., Gediga, G.: Rough set data analysis: A road to non-invasive knowledge discovery. Methodos Publishers, Bangor (2000)

Dumitrescu, A., Mitchell, J.S.B.: Approximation algorithms for tsp with neighborhoods in the plane. In: SODA '01: Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms, pp. 38–46 (2001)

Dumont, B., Boissy, A., Achard, C., Sibbald, A.M., Erhard, H.W.: Consistency of animal order in spontaneous group movements allows the measurement of leadership in a group of grazing heifers. Appl. Anim. Behav. Sci. **95**(1–2), 55–66 (2005)

Dunn, J.: Intergraph at the Heights. Business Alabama **18**(8), 19–24 (2003)

Düntsch, I., Gediga, G.: Uncertainty measures of rough set predictions. Artif. Intell. **106**, 109–137 (1998)

Durand, F.: An invitation to discuss computer fepiction. In symposium on non-photorealistic animation and rendering (NPAR) (2002)

Dykes, J.: Cartographic Visualization: Exploratory Spatial Data Analysis with Local Indicators of Spatial Association Using Tcl/Tk and cdv. Statistician **47**, 485–497 (1998)

Dykes, J.: Exploring Spatial Data Representation with Dynamic Graphics, Computers and Geosciences. **23** 345–370 (1996)

Dykes, J., MacEachren, A.M., Kraak, M.J. (eds.): Exploring Geovisualization. Elsevier, Amsterdam (2005)

Dykes, J., MacEachren, A.M., Kraak, M.-J.: Exploring Geovisualization. Elsevier, London (2004)

Dykes, J.A., Mountain, D.M.: Seeking structure in records of spatio-temporal behaviour: visualization issues, efforts and application. Comput. Stat. Data Anal. **43**(4), 581–603 (2003)

Easterfield, M.E., Newell, R.G., Theriault, D.G.: Version Management in GIS Applications and Techniques, EGIS '90 Conference Proceedings, Amsterdam, April (1990)

Ebdon, D.: Statistics in Geography (2nd edn. with corrections). Blackwell, Oxford (1988)

Echardt, K., Ulbrich, U.: Potential impacts of climate change on groundwater recharge and streamflow in a central European low mountain range. J. Hydrol. **284**(1–4), 244–252 (2003)

Eck, J., Chainey, S., Cameron, J., Leitner, M., Wilson, R.E.: Mapping Crime: Understanding Hot Spots. Mapping and Analysis for Public Safety, National Institute of Justice (2005)

Eck, J., Wartell, J.: Reducing Crime and Drug Dealing by Improving Place Management: A Randomized Experiment. National Institute of Justice (1997)

ECS, Release 6B Implementation Earth Science Data Model for the ECS Project. http://spg.gsfc.nasa.gov/standards/heritage/eosdis-core-system-data-model. Accessed 13 Sept 2007

Edmonds, J., Johnson, E.L.: Matching, Euler tours and Chinese Postman, Math. Program. **5**, 88–124 (1973)

Edwardes, A., Burghardt, D., Dias, E., Purves, R.S., Weibel, R.: Geo-enabling spatially relevant data for mobile information use and visualisation. In: Proceedings of the 5th International Workshop on Web and Wireless GIS, W2GIS 2005. Lecture Notes in Computer Science, vol. 3833, pp. 78–92. Springer-Verlag, Berlin, Heidelberg (2005)

Edwards Jr., T.C., Homer, C.G., Bassett, S.C., Falconer, A., Ramsey, R.D., Wight, D.W.: Utah Gap analysis: an environmental information system. Final Project Report 95–1. Logan: Utah Cooperative Fish and Wildlife Research Unit (1995)

Egenhofer, M. (2002) Toward the Semantic Geospatial Web. In: The Tenth ACM International Symposium on Advances in Geographic Information Systems. Available via. http://www.spatial.maine.edu/~max/SGW02.pdf. Accessed 8-9 Nov 2002

Egenhofer, M.: Deriving the composition of binary topological relations. J. Vis. Lang. Comput. **5**(2), 133–149 (1994)

Egenhofer, M.: A Formal Definition of Binary Topological Relationships. In: Litwin, W., Schek, H.J. (eds.) Proc. 3rd Intl. Conf. on Foundations of Data Organization and Algorithms (FODO). Lecture Notes in Computer Science 367. pp. 457–472. Springer Verlag, Paris (1989)

Egenhofer, M.J., Al-Taha, K.: Reasoning about gradual changes of topological relationships. In: Frank, A.U., Campari, I., Formentini, U. (eds.) Theories and Models of Spatio-temporal Reasoning in Geographic Space. Lecture Notes in Computer Science, vol. 639, pp. 196–219. Springer, Berlin (1992)

Egenhofer, M.J., Clementini, E., di Felice, P.A.: Topological relations between regions with holes. International Journal of Geographical Information Systems **8**(2), 129–142 (1994)

Egenhofer, M.J., Franzosa, R.D.: Point set topological relations. Int. J. Geogr. Inf. Syst. **5**, 161–174 (1991)

Egenhofer, M.J., Golledge, R.G. (eds.): Spatial and Temporal Reasoning in Geographic Information Systems. Oxford University Press, New York (1998)

Egenhofer, M.J., Herring, J.: A mathematical framework for the definition of topological relationships. In: Proceedings of the Fourth International Symposium on Spatial Data Handling, pp. 803–813. Zürich, Switzerland (1990)

Egenhofer, M.J., Herring, J.R.: Categorizing binary topological relationships between regions, lines, and points in geographic databases. Department of Surveying Engineering, University of Maine, Orono, ME (1991)

Egenhofer, M.J., Sharma, J.: Topological relations between regions in R^2 and Z^2. In: David, A., Ooi, B. (eds.): Proceedings of the 3rd International Symposium on Advances in Spatial Databases, SSD'93, volume 692 of LNCS, Springer, Heidelberg (1993)

Egenhofer, M.J.: Toward the semantic geospatial web. In: Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems, McLean, Virginia, 8–9 November 2002

Egenhofer, M.: Naïve geography. In: Frank, A., Kuhn, W. (eds.) COSIT '95, Semmering, Austria, September 1995. Lecture Notes in Computer Science, vol. 988, pp. 1–15, Springer-Verlag, (1995)

Egenhofer, M., Sharma, J., Mark, D.: A Critical Comparison of the 4-Intersection and 9-Intersection Models for Spatial Relations: Formal Analysis. In: McMaster, R., Armstrong M. (eds.) Proceedings of AutoCarto 11 Minneapolis (1993)

Egenhofer, M.: Toward the Semantic Geospatial Web. Voisard, A., Chen, S.C. (eds.) ACM-GIS 2002. pp. 1–4. McLean, Virginia, USA (2002)

Egger, O., Fleury, P., Ebrahimi, T., Kunt, M.: High-performance compression of visual information-a tutorial review-part I: still pictures. In: Proceedings IEEE, vol. 87, pp. 976–1011. June 1999

Ehlschlaeger, C.R., Shortridge, A.M., Goodchild, M.F.: Visualizing spatial data uncertainty using animation. Comput. & Geosci. **23**, 387–395 (1997)

Eidenbenz, C., Kaser, C., Baltsavias, E.: ATOMI – Automated Reconstruction of Topographic Objects from Aerial Images using Vectorized Map Information. Int. Arch. Photogramm. Remote Sens. **33**(Part 3/1), 462–471 (2000)

EKahau: [Online]. Available: http://www.ekahau.com/ (2006)

Elbassioni, K., Elmasry, A., Kamel, I.: An efficient indexing scheme for multi-dimensional moving objects. In: Proceedings of the 9th Intern. Conf. ICDT, Siena, Italy, 8–10 2003, pp. 425–439

Elbassioni, K., Elmasry, A., Kamel, I.: An indexing method for answering queries on moving objects. Distrib. Parallel Databases **17**(3), 215–249 (2005)

Elias, B., Brenner, C.: Automatic generation and application of landmarks in navigation data sets. Paper presented at Developments in Spatial Data Handling, Proceedings of the 11th International Symposium on Spatial Data Handling, Leicester, UK, 2004

Elias, B.: Extracting Landmarks with Data Mining Methods. In: Kuhn, W., Worboys, M., Timpf, S. (eds.) Spatial Information Theory: Foundations of Geographic Information Science, pp. 375–389. Springer-Verlag, Berlin (2003)

Elliott, P., Wakefield, J.C., Best, N.G., Briggs, D.J. (eds.): Spatial Epidemiology: Methods and Applications. Oxford University Press, London (2000)

Elmasri, R., Navathe, S.B.: Fundamentals of Database Systems, 3rd edn. Addison-Wesley, Reading, MA (2000)

Elmasri, R., Navathe, S.B.: Fundamentals of Database Systems (5th edn.), Addison Wesley, Boston, MA, USA, March (2006) ISBN-13: 978-0321369574

Elmasri, R., Navathe, S.: Fundamental of Database Systems (5. edn). Addison Wesley, Boston, MA, USA (2007)

EM-DAT Emergency Disasters Database. http://www.em-dat.net/links/disasterdbs.html. Accesed 20 Aug 2007

Emerson, C.W., Quattrochi, D.A., Lam, N.S.: Spatial metadata for remote sensing imagery. In: Proceedings of the 4th Annual Earth Science Technology Conference (ESTC), Palo Alto, CA, USA (2004)

Environmental Protection Agency (EPA), Better Assessment Science Integrating Point and Nonpoint Sources (BASINS): User's Manual, Version 3.0, U.S. EPA Office of Water (4305), Washington, D.C., EPA-823-H-01-001, http://www.Epa.gov/ost/basins (2001)

Erl, T.: Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall, Upper Saddle River, USA (2005)

Erwig, M., Güting, R.H., Schneider, M., Vazirgiannis, M.: Spatio-temporal Data Types: An approach to modeling and querying moving objects in databases. GeoInformatica **3**, 265–291 (1999)

Erwig, M., Güting, R., Schneider, M.,Vazirgiannis, M.: Spatio-temporal data types: An approach to modeling and querying moving objects in databases. ChoroChronos Research Project, Technical Report CH-97–08 (1997)

Erwig, M., Schneider, M.: Developments in spatio-temporal query languages. IEEE Int. Workshop on Spatio-temporal Data Models and Languages (STDML), pp. 441–449 (1999)

Erwig, M., Schneider, M.: Spatio-temporal predicates. IEEE Trans. Knowledge Data Eng. **14**, 881–901 (2002)

Erwig, M., Schneider, M.: Vague Regions. 5th Int. Symp. on Advances in Spatial Databases, pp. 298–320. Springer-Verlag, Berlin, Heidelberg, Germany (1997)

Erwig, M., Schneider, M.: Vague regions. In: Scholl, M., Voisard, A. (eds.) Advances in Spatial Databases – Fifth International Symposium, SSD '97. Lecture Notes in Computer Science, vol. 1262, pp. 298–320. Springer, Berlin (1997)

ESA – Observing the Earth. http://www.esa.int/esaEO/index.html. Accessed November 2006

ESG (2006) Earth System Grid. https://www.earthsystemgrid.org/home/publicHomePage.do. Accessed 10 Sept 2006

Espenshade, E.B. Jr., et al. (eds.): Goode's World Atlas. 19th edn. Rand McNally, Skokie, Illinois, USA (1995)

ESRI: ArcGIS. http://www.esri.com/software/arcgis/index.html

ESRI. ArcIMS. http://www.esri.com/software/arcgis/arcims/index.html. Accessed 05 March 2007

ESRI: ArcXML Programmer's Reference Guide, ArcIMS 4. (2002)

ESRI web site: http://www.esri.com

ESRI: What is ArcGIS 9.2. Redlands, California (2006)

Ester, M., Kriegel, H.P., Sander, J.: Algorithms and applications for spatial data mining. Geographic Data Mining and Knowledge Discovery, Research Monographs. In: GIS Chapter 7 (2001)

Ester, M., Kriegel, H., Sander, J.: Spatial data mining: A database approach. In: 5th International Symposium on Advances in Spatial Databases, pp 47–66. Springer-Verlag, London, UK (1997)

Eubank, S., Anil Kumar, V.S., Marathe, M., Srinivasan, A., Wang, N.: Structure of Social Contact Networks and their Impact on Epidemics. In: AMS-DIMACS Special Volume on Epidemiology **70**, 181–213 (2005)

Eubank, S., Guclu, H., Anil Kumar, V.S., Marathe, M., Srinivasan, A., Toroczkai, Z., Wang, N.: Modeling Disease Outbreaks in Realistic Urban Social Networks. Nature **429**, 180–184 (2004)

Evans, J., Clarence, F.: Deploying IP and MPLS QoS for Multiservice Networks: Theory and Practice. Morgan Kaufmann, San Francisco, CA (2007)

Evans J (ed) (2002) Web Coverage Service (WCS) Version 1.0.0. OpenGIS Project Document 03-065r6. https://portal.opengeospatial.org/files/?artifact_id=3837

Evans, J. (ed.): OGC, Web coverage service (WCS), Version 1.0.0. Wayland, Massachusetts (2005)

Evans, J. (eds.) (2003) OpenGIS Web Coverage Service(WCS) Implementation Specification. OGC doi:03-065r6. https://portal.opengeospatial.org/files/?artifact_id=3837

Evans, J.R., Minieka, E.: Optimization Algorithms for Networks and Graphs. Marcel Dekker Inc., New York (1992)

Evenden, G.: Brief History of PROJ[4] (2004) http://lists.maptools.org/pipermail/proj/2004-March/001117.html. Accessed 17 Feb 2007

Evennou, F., Marx, F., Novakov, E.: Map-aided indoor mobile positioning system using particlefilter. In: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, USA, vol. 4, pp. 2490–2494. IEEE, Piscataway, NJ (2005)

Everitt, B.S., Landau, S., Leese, M.: Cluster Analysis, 4th edn. Oxford University Press, New York (2001)

Everly B: (2002) Personal communication. Product support, help@neuralware.com. Contacted 03.18.02

For example, two such conferences include the Eighth International Conference on Computers in Urban Planning and Urban Management cited in [35] and held in Sendai, Japan, in 2003; and the Third Annual Public Participation Geographic Information Science (PPGIS) Conference held in Madison, WI, in 2004; the latter was co-sponsored by the Urban and Regional Information Systems Association (URISA) and the National Consortium for Rural Geospatial Innovations (www.ruralgis.org) – this conference was included in the citation for [13]

Executive Office of the President (EOP) (1994) Coordinating geographic data acquisition and access: the National Spatial Data Infrastructure. http://www.fgdc.gov/publications/documents/geninfo/execord.html. Accessed 20 Aug 2007

Executive order 12906 (1994) http://www.fgdc.gov/policyandplanning/excecutive_order. Accessed 9 Sept 2007

Fagin, R.: Combining fuzzy information from multiple systems (extended abstract). In: PODS 96: Proceedings of the fifteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, pp. 216–226, Montreal, Canada, 3–5 June 1996

Fagin, R., Kolatis, P.G., Popa, L., Tan, W.C.: Composing schema mappings: second-order dependencies to the rescue: In: Symposium on Principles of Database Systems (PODS) 2004, June 14–16, Paris, France

Faiz, S.: Managing geographic data quality during spatial data mining and spatial olap. data warehousing and data quality. GIM Int **14**(12), 28–31 (2000)

Faloutsos, C., Equitz, W., Flickner, M., Niblack, W., Petkovic, D., Barber, R.: Efficient and Effective Querying by Image Content. J. Intell. Inf. Syst. **3**(3), 231–262 (1994)

Faloutsos, C.: Gray codes for partial match and range queries. IEEE Trans. Soft. Eng., TSE **14**(10), 1381–1393 (1988)

Faloutsos, C., Jagadish, H.V., Mendelzon, A., Milo, T.: Signature technique for similarity-based queries. In Proc. of SEQUENCES (1997)

Faloutsos, C., Kamel, I.: Beyond uniformity and independence: analysis of r-trees using the concept of fractal dimension. In: Proc. of ACM Symposium on Principles of Database Systems (PODS), Minnesota, USA, 24–26 May 1994, pp. 4–13

Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time-series databases. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, Minneapolis, 25–27 May 1994, pp. 419–429

Faloutsos, C., Rong, Y.: Dot: A spatial access method using fractals. In: Proc. of Intl. Conf. on Data Engineering, ICDE, Kobe, Japan, April 1991, pp. 152–159

Faloutsos, C.: Searching Multimedia Databases by Content. Kluwer Academic Publishers, Norwell, MA, USA (1996)

Faloutsos, C., Sellis, T., Roussopoulos, N.: Analysis of object oriented spatial access methods. SIGMOD Record **16**(3):426–439 (1987)

Fang, M., Shivakumar, N., Garcia-Molina, H., Motwani, R., Ullman, J. D.: Computing iceberg queries efficiently. In: Proc. of VLDB (1998)

Farradyne, P.B.: A Guide to Updating Highway Emergency Response Plans for Terrorist Incidents. The American Association of State Highway and Transportation Officials' Security Task Force (2002)

Farrington, P., Andrews, N.: Outbreak detection: Application to infectious disease surveillance. In: Brookmeyer, R., Stroup, D. (eds.): Monitoring the Health of Populations: Statistical Principles and

Methods for Public Health Surveillance. chapter 8. Oxford University Press, London (2004)

Faugeras, O.D., Luong, Q.T., Papadopoulo, T.: The Geometry of Multiple Images. MIT Press, Cambridge, MA (2001)

Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery: An overview. In: Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Ulthurusamy, R. (eds.) Advances in Knowledge Discovery and Data Mining, pp. 1–34. MIT Press, Cambridge, MA (1996)

FCC Docket No. 94–102: Revision of the commissions rules to insure compatibility with enhanced 911 emergency calling systems. Federal Communications Commission Tech Rep RM-8143, Julyyes (1996)

Federal Geographic Data Committee (1998) Geospatial Metadata Standards. http://www.fgdc.gov/metadata/geospatial-metadata-standards. Cited 23 Mar 2007

Federal Geographic Data Committee: Business Case for Metadata. http://www.fgdc.gov/metadata/metadata-business-case (2005)

Federal Geographic Data Committee: Components of the National Spatial Data Infrastructure. Information on Geospatial Standards. http://www.fgdc.gov, viewed on November 26 (2006)

Federal Geographic Data Committee: Content Standard for Digital Geospatial Metadata. http://www.fgdc.gov/standards/projects/FGDC-standards-projects/metadata/base-metadata/index_html (2006)

Federal Geographic Data Committee: Content Standard for Digital Geospatial Metadata Workbook. http://www.fgdc.gov/metadata/metadata-publications-list (2000)

Federal Geographic Data Committee: Geospatial Metadata Factsheet. http://www.fgdc.gov/metadata/metadata-publications-list (2005)

Federal Geographic Data Committee: Geospatial Metadata. http://www.fgdc.gov/metadata (2006)

Federal Geographic Data Committee: Geospatial Metadata Quick guide. http://www.fgdc.gov/metadata/metadata-publications-list (2005)

Federal Geographic Data Committee and the National Metadata Cadre: Top Ten Metadata Errors. http://www.fgdc.gov/metadata/metadata-publications-list (2006)

Federal Geographic Data Committee, www.fgdc.gov The FGDC website provides extensive resources about the NSDI concepts discussed in this entry

Feeman, T.G.: Portraits of the Earth: A Mathematician Looks at Maps. American Mathematical Society, Providence (2002)

Feick, R.D., Hall, B.G.: A method for examining the spatial dimension of multicriteria weight sensitivity. Int J Geogr Inf Sci **20**(7), 703–726 (2004)

Feiner, S., McKeown, K.: Automating the generation of coordinated multimedia. IEEE Computer **24**(10), 33–41 (1994)

Fekete, S., Meijer, H., Rohe, A., Tietze, W.: Solving a Hard problem to approximate an Easy one: heuristics for maximum matchings and maximum traveling salesman problems. J. Exp. Algorithm. (JEA)**7**, (2002)

Fellbaum, C. (ed.): Wordnet: An Electronic Lexical Database. MIT Press, Cambridge, MA (1998)

Felson, M.: Crime and Everyday Life. Pine Forge (1994)

FEMA: FEMA History. http://www.fema.gov/about/history.shtm. Accessed 10 Sept 2007

Fensel, D., Wahlster, W., Lieberman, L., Hendler, J.: Spinning the Semantic Web: Bringing the World Wide Web to its Full Potential. MIT Press, Cambridge, MA (2002)

Ferber, J.: Multi-Agent Systems – An Introduction to Distributed Artificial Intelligence. Addison-Wesley, London, UK (1999)

Ferguson, N.L., Cummings, D.A.T., Cauchemez, S., Fraser, C., Riley, S., Meeyai, A., Lamsirithaworn, S., Burke, D.S.: Strategies for containing an emerging influenza pandemic in Southeast Asia, Nature **437**, 209–214 (2005)

Ferguson, N.L., Cummings, D.A.T., Fraser, C., Cajka, J.C., Cooley, P.C., Burke, D.S.: Strategies for mitigating an influenza pandemic, Nature, 448–452 (2006)

Ferhatosmanoglu, H., Agrawal, D., Abbadi, A.E.: Concentric hyperspaces and disk allocation for fast parallel range searching. ICDE 608–615 (1999)

Ferhatosmanoglu, H., Tuncel, E., Agrawal, D., El Abbadi, A.: Vector approximation based indexing for non-uniform high dimensional data sets. In: CIKM '00: Proceedings of the ninth international conference on Information and knowledge management, pp. 202–209. ACM Press, New York, NY (2000)

Ferraiolo, J., Fujisawa, J., Jackson, D.: Scalable Vector Graphics (SVG) 1.1 Specification (2003), http://www.w3.org/TR/SVG11/index.html, accessed 2006–11–19

FGDC (1994) This website has information for clearinghouse, NSDI, GOS portal, and funding opportunities. http://www.fgdc.gov/. Accessed 20 Aug 2007

FGDC (1998) Content Standard for Digital Geospatial Metadata (version 2.0), FGDC-STD-001-1998. Reston, VA. http://www.fgdc.gov/standards/projects/FGDC-standards-projects/metadata/base-metadata/index_html. Accessed 9 Sept 2007

FGDC (1999), Content Standard for Digital Orthoimagery, FGDC-STD-008-1999. Reston, VA. http://www.fgdc.gov/standards/projects/FGDC-standards-projects/orthoimagery/index_html. Accessed 9 Sept 2007

FGDC (1999), Content Standard for Remote Sensing Swath Data, FGDC-STD-009-1999. http://www.fgdc.gov/standards/projects/FGDC-standards-projects/swath_data/index_html. Accessed 9 Sept 2007

FGDC (2002), Content Standard for Digital Geospatial Metadata: Extensions for Remote Sensing Metadata, FGDC-STD-012-2002. http://www.fgdc.gov/standards/projects/FGDC--standard-projects/csdgm_rs_ex/rermote-sensing-metadata. Accessed 9 Sept 2007

FGDC (2004). NSDI Futurer Directions Initiative. http://www.fgdc.gov/policyandplanning/future-directions/reports/FD_Final_Report.pdf. Accessed 29 Apr 2007

FGDC Clearinghouse: Node performance measurement (1999) http://www.fgdc.gov/clearinghouse. Accessed 20 Jul 2005

FGDC, CSDGM. http://www.fgdc.gov/standards/standards_publications/index_html. Accessed 13 Sept 2007

FGDC, The Federal Geographic Data Committee Homepage. http://www.fgdc.gov. Accessed 1 May 2007

FGDC: Framework, Introduction and GuideBook. Federal Geographic Data Committee, Washington (1997)

FGDC Standard Working Group homepage. http://www.fgdc.gov/standards/organization/FGDC-SWG/index_html. Accessed 9 Sept 2007

FHWA: TSIS Manuals and Documentation. FHWA, Washington, D.C. (2003)

Fielden, N.: History of Information Retrieval Systems & Increase of Information Over Time. In: Biennial California Academic & Research Librarians (CARL) Conference, Monterey, CA, US 10 May 2002

Figueira, J., Greco, S., Ehrgott, M.: Multiple Criteria Decision Analysis: State of the Art Surveys. Springer-Verlag, New York (2005)

Filho, J.L., Sodre, V.D.F., Daltio, J., Rodrigues Junior, M.F., Vilela, V.: A CASE Tool for Geographic Database Design Supporting

Analysis Patterns. Lecture Notes in Computer Science, vol. 3289, pp. 43–54. Springer-Verlag, Berlin/Heidelberg (2004)

Filin, S., Doytsher, Y.: A Linear Conflation Approach for the Integration of Photogrammetric Information and GIS Data. Int. Arch. Photogramm. Remote Sens. **33**, 282–288 (2000)

Fink, C.N.: Anti-Terrorism Security and Surface Transportation Systems: A Review of Case Studies and Current Tactics. Presented at Annual Meeting of Transportation Research Board, Washington, D.C., Jan 2003

Finkel, R.A., Bentley, J.L.: Quad Trees, a data structure for retrieval on composite keys. Acta Informatica **4**(1):1–9 (1974)

Finke, R.A., Ward, T.B., Smith, S.M.: Creative Cognition: Theory, Research, and Applications. The MIT Press, Cambridge, MA (1992)

Finley, A.O., Banerjee, S., Carlin, B.P.: spBayes: An R package for univariate and multivariate hierarchical point-referenced spatial models. J. Statistical Softw. **19**, 4 (2007)

Fisher, P., Dykes, J., Wood, J.: Map Design and Visualization. The Cartographic Journal. **30** 136–142 (1993)

Fisher, P.: Editorial, Citations to the International Journal of Geographical Information Systems and Science: the first 10 years. Int. J. Geogr. Inf. Sci. **15**, 1–6 (2001)

Fisher, P.E., Tate, N.J.: Causes and consequences of error in digital elevation models. Prog. Phys. Geogr. **30**, 467–489 (2006)

Fisher, P.F.: Algorithm and implementation uncertainty in viewshed analysis. Int. J. Geogr. Inf. Syst. **7**, 331–347 (1993)

Fisher, P.F., Cheng, T., Wood, J.: Higher order vagueness in geographical information: Empirical geographical population of Type n fuzzy sets. GeoInformatica (in press)

Fisher, P.F.: Models of Uncertainty in Spatial Data. In: Longley, P., Goodchild, M., Maguire, D., Rhind, D. (eds.) Geographical Information Systems: Principles Techniques Management and Applications, vol. 1, pp. 191–205. Wiley and Sons, New York (1999)

Fisher, P.F.: The pixel: a snare and a delusion. Int. J. Remote Sens. **18**, 679–685 (1997)

Fisher, P.F., Unwin, D.J. (eds.): Re-Presenting GIS. Wiley, Hoboken, NJ (2005)

Fisher, P.F., Wood, J., Cheng, T.: Where is Helvellyn? Multiscale morphometry and the mountains of the English Lake District. Trans. Inst. British Geogr. **29**, 106–128 (2004)

Fitzke, J., Greve, K., Müller, M., Poth, A.. Building SDIs with Free Software – the deegree project. In: Proceedings of CORP 2004, Vienna, Austria, 25–17 February, 2004. Available via http://www.corp.at

Fix E., Hodges, J.L.: Discriminatory analysis: non-parametric discrimination: consistency properties, Report No. 4. USAF School of Aviation Medicine, Randoph Field (1951)

Fix E., Hodges, J.L.: Discriminatory analysis: non-parametric discrimination: small sample performance, Report No. 11. USAF School of Aviation Medicine, Randolph Field (1952)

Flavie, M., Fortier, A., Ziou, D., Armenakis, C., Wang, S.: Automated Updating of Road Information from Aerial Images. In: Proceedings of American Society Photogrammetry and Remote Sensing Conference. Amsterdam, Holland (2000)

Fleischmann, B., Sandvoß, E., Gnutzmann, S.: Dynamic vehicle routing based on on-line traffic information. Trans. Sci. **38**(4), 420–433 (2004)

Fleischner, H.: (Some of) the many uses of Eulerian Graphs in graph theory, Discrete Math. **230**, 23–43 (2001)

Flüeler, I., Iosifescu-Enescu, I., Neumann, A.: Cartographic SVG Applications as Risk Management Support and Communication Platforms. In: Proceedings of the Workshop "Visualization, Analytics & Spatial Decision Support" at GIScience 2006 in Münster, Germany (2006)

Foerster, T., Stoter, J.: Establishing an OGC web processing service for generalization processes. Paper presented at the ICA Workshop on Generalization and Multiple Representation, Portland, 25 June 2006. Available via: http://ica.ign.fr/Portland/paper/ICA2006-foerster_stoter.pdf Accessed 12 Feb 2007

Fohere, N., Haverkamp, S., Frede, H.G.: Assissment of the effects of land use patterns on hydrologic landscape functions: development of sustainable land use concepts for low mountain range areas. Hydrol. Process. **19**(3), 659–672 (2005)

Fohl, P., Curtin, K.M., Goodchild, M.F., Church, R.L.: A Non-Planar, Lane-Based Navigable Data Model for ITS, In: Proc. International Symposium on Spatial Data Handling, pp. 17–29 (1996)

Foley, K.M., Fuentes, M.: A statistical framework to combine multivariate spatial data and physical models for hurricane surface wind prediction. JABES, to appear

Follin, J.-M., Bouju, A., Bertrand, F., Boursier, P.: Multi-resolution extension for transmission of geodata in a mobile context. Comput. Geosci. **31**, 179–188 (2005)

Fonseca, F., Egenhofer, M., Agouris, P., Câmara, G.: Using Ontologies for Integrated Geographic Information Systems. Transactions in GIS. **6**(3), 231–257 (2002)

Fonseca, F., Egenhofer, M., Davis, C., Camara, G.: Semantic granularity in ontology-driven geographic information systems. Ann. Math. Artif. Intell. **36**(1–2), 121–151 (2002)

Fonseca, F., Egenhofer, M.: Ontology-Driven Geographic Information Systems. In: 7th ACM Symposium on Advances in Geographic Information Systems, Kansas City, MO, pp. 14–19. (1999)

Fonseca, F.: From Geo-Pragmatics to Derivation Ontologies: New Directions for the GeoSpatial Semantic Web. Transactions in GIS. **11**(3), 313–316 (2007)

Fonseca, F., Sheth, A.: (2002) The geospatial semantic web. UCGIS White Paper. http://www.ucgis4.org/priorities/research/2002researchagenda.htm

Fonseca, F., Sheth, A. (2002) The Geospatial Semantic Web. University Consortium for Geographic Information Science. http://www.ucgis.org/priorities/research/2002researchPDF/shortterm/e_geosemantic_web.pdf. Accessed 21 March 2007

Foody, G.M., Atkinson, P.M. (eds.): Uncertainty in Remote Sensing and GIS. Wiley, Chicester (2002)

Forlizzi, L., Guting, R.H., Nardelli, E., Schneider, M.: A Data Model and Data Structures for Moving Objects Databases. In: Proceedings of ACM-SIGMOD International Conference on Management of Data, pp. 319–330. Dallas, Texas, USA (2000)

Forman S, Steen L (2006) Global Positioning System. http://www.stolaf.edu/other/ate/gps.html. Accessed 8 May 2006

ForNet: Internet delivery of spatial and related data. http://www.gis.umn.edu/

Foschi, P. G., Kollipakkam, D., Liu, H., Mandvikar, A.: Feature extraction for image mining (2007) www.public.asu.edu/~huanliu/papers/mis02.pdf. Accessed on 01.10.07

Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The physiology of the grid: an open grid services architecture for distributed systems integration. Open Grid Service Infrastructure WG, Global Grid Forum, 22 June 2002

Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the grid: enabling scalable virtual organizations. Int. J. Supercomput. Appl. **15**, 200–222 (2001)

Fotheringham, A.S., Brunsdon, C., Charlton, M.: Geographically Weigthed Regression the Analysis of Spatially Varying Relationships. Wiley, West Sussex, UK (2002)

Fotheringham, A.S., Brunsdon, C., Charlton, M.: Quantitative Geography Perspectives on Spatial Data Analysis. Sage, London, UK (2000)

Fotheringham, A.S., O'Kelly, M.E.: Spatial Interaction Models: Formulations and Applications. Kluwer, Boston (1989)

Fowler, M.: UML 2.0, p. 165. Campus Press, Pearson Education, Paris (2004)

Fox, D., Hightower, J., Kauz, H., Liao, L., Patterson, D.: Bayesian Techniques for Location Estimation. In: Proceedings of the Workshop on Location-Aware Computing (UBICOMP Conference), Seattle, WA, October 2003

Francica, J.: LocationBI – Location-based Business Intelligence, Directions Magazine, October 22 (2003)

Frankel, D.S., Harmon, P., Mukerji, J., Odell, J., Owen, M., Rivitt, P., Rosen, M., Soley, R.: The Zachman Framework and the OMG's Model Driven Architecture. Business Process Trends, Whitepaper (2003)

Frankenberger, J.R., Brooks, E.S., Walter, M.T., Walter, M.F., Steenhuis, T.S.: A GIS-based variable source area hydrology model. Hydrol. Process **13**, 805–822 (1999)

Frank, A., Grumbach, S., Guting, R.H., Jensen, C.S., Koubarakis, M., Lorentzos, N., Manolopoulos, Y., Nardelli, E., Pernici, B., H.-J. Schek, Scholl, M., Sellis, T., Theodoulidis, B., Widmayer, P.: Chorochronos: a research network for spatiotemporal database systems. SIGMOD Rec. **28**, 12–21 (1999)

Frank, A.: Qualitative spatial reasoning about distances and directions in geographic space. J. Visual Lang. Comput. **3**, 343–371 (1992)

Frank, A.: Spatial Communication with Maps: Defining the Correctness of Maps Using a Multi-Agent Simulation. In: Freksa, C., et al. (eds.) Spatial Cognition II – Integrating Abstract Theories, Empirical Studies, Formal Methods, and Practical Applications, p. 80–99. Springer, Berlin, Heidelberg, Germany (2000)

Frank, A.U.: Metamodels for data quality description. In: Jeansoulin, R., Goodchild, M. (eds.) Data Quality in Geographic Information – From Error to Uncertainty, pp. 15–29. Editions Hermès, Paris (1998)

Frank, A.U.: Ontology for spatio-temporal databases. In: Koubarakis, M., Sellis, T.K., Frank, A.U. (eds.) Spatiotemporal Databases: the Chorochronos Approach, pp. 9–77. Springer-Verlag, Berlin (2003)

Frank, A.U.: Tiers of ontology and consistency constraints in geographic information systems. Int. J. Geogr. Inf. Sci. **15**, 667–678 (2001)

Frank, W.: Gdal – geospatial data abstraction library. http://www.remotesensing.org/gdal/

Franzese: Oak Ridge Evacuation Modeling System (OREMS), User's Guide. Oak Ridge National Laboratory. Oak Ridge, Tennessee (2005)

Franzese, O., Sorensen, J.: Fast Deployable System for Consequence Management: The Emergency Evacuation Component. In: Proceedings of the 2004 ITS Safety and Security Conference, Miami Beach, FL (2004)

Franz, W., Eberhardt, R., Luckenbach, T.: FleetNet – Internet on the road, The 8th World Congress on Intelligent Transportation Systems, Sydney, Australia, Oct 2001

Frederickson, G.N.: Fast algorithms for shortest paths in planar graphs, with applications. In: SIAM J. Computing 16(6), pp. 1004–1022 (1987)

Fredkin, E.: Trie memory. Communications of the ACM **3**(9):490–499 (1960)

Free Software Foundation (1991) GNU General Public License. http://www.gnu.org/copyleft/gpl.html. Accessed 1 Jan 2007

Freksa, C.: Temporal reasoning based on semi-intervals. Artif Intell **54**, 199–227 (1992)

Freksa, C.: Using orientation information for qualitative spatial reasoning. In: Andrew U.F., Campari, I., Formentini, U. (eds.) Theories and Methods of Spatio-temporal Reasoning. Lecture Notes Computing Science vol. 639, pp. 162–178. Springer, Berlin (1992)

Frigeri, A., Federico, C., Pauselli, C., Minelli, G.: Identifying wrinkle ridges from mars mgs and viking mission data: Using grass GIS in planetary geology. Trans. GIS **8**(2), 255–262 (2004)

Frisén, M., Sonesson, C.: Optimal surveillance. In: Lawson, A.B., Kleinman, K. (eds.) Spatial and Syndromic Surveillance for Public Health, chapter 3. Wiley, New York (2005)

Frisen, M.: Statistical surveillance: optimality and methods. Int. Stat. Rev.. **71**, 1403–1434 (2003)

Frisken, S., Perry, R.: Simple and Efficient Traversal Methods for Quadtrees and Octrees. http://www.merl.com/reports/docs/TR2002-41.pdf

Fritz, S., See, L.: Comparison of land cover maps using fuzzy agreement. Int. J. Geogr. Inf. Sci. **19**, 787–807 (2005)

Fröhlich, C., Mettenleiter, M.: Terrestrial laser scanning—new perspectives in 3D surveying. Int. Arch. Photogramm. Remote Sensing Spatial Inf. Sci. **36**, 7–13 (2004). Available via http://www.isprs.org/commission8/workshop_laser_forest/FROEHLICH.pdf (2007). Accessed 6 July 2007

Frolov, Y.S., Maling, D.H., The accuracy of area measurement by point counting techniques. Cartogr. J. **1**, 21–35 (1969)

Fryer, J.G., Mitchell, H.L., Chandler, J.H.: Applications of 3D Measurement From Images. Whittles, Caithness, UK (2006)

Frysinger, S., Copperman, D.A., Levantino, J.P.:. Environmental decision support systems (EDSS): an open architecture integrating modeling and GIS. In: Goodchild, M.F., Steyaert, L.T., Park, B.O., Johnston, C., Maidment, D., Crane, M., Glendinning, S. (eds.) GIS and Environmental Modeling: Progress and reasearch Issues, pp. 357–362. GIS World Books, Fort Collins, CO

Fuentes, M., Chen, L., Davis, J.M., Lackmann, G.M.: Modeling and predicting complex space-time structures and patterns of coastal wind fields. Environmetrics **16**, 449–464 (2005)

Fuentes, M.: A high frequency kriging approach for non-stationary environmental processes. Environmetrics **12**, 469–483 (2001)

Fuentes, M., Raftery, A.E.: Model evaluation and spatial interpolation by Bayesian combination of observations with outputs from numerical models. Biom. **61**, 36–45 (2005)

Fuller, G.W.: Capabilities for a Future National Intelligence Enterprise (2005)http://www.garywfuller.net/pubs/capabilities_nie.html. Accessed 29 Aug 2007

Fulton, N.: High-tech cloning. http://blogs.reuters.com/2006/07/22/high-tech-cloning/, Reuters: Retreived 10/10/2006 (2006)

Fung, W.F., Sun, D., and J. Gehrke, J.: COUGAR: The Network is the Database. In: Proceedings of International ACM SIG Conference Modeling of Data (SIGMOD), Madison, WI 2002

Furlanello, C., Neteler, M., Merler, S., Menegon, S., Fontanari, S., Donini, A., Rizzoli, A., Chemini, C.: GIS and the Random Forest Predictor: Integration in R for Tick-borne Disease Risk Assessment. In: Hornik, K., Leisch, F. (eds.) Proceedings of the 3rd International Workshop on Distributed Statistical Computing, March 20–22, Vienna, Austria (2003)

Gaede, V., Gunther, O.: Multidimensional Access Methods. ACM Comput. Surv. **30**(2), 170–231 (1998)

Gahegan, M., Hardisty, F., Takatsuka, M., Wheeler, M.: Introducing Geo VISTA Studio: an integrated suite of visualization and computational methods for exploration and knowledge construction in geography. Comput. Environ. Urban Syst. **26**, 267–292 (2002)

Gahegan, M.: On the application of inductive machine learning tools to geographical analysis. Geogr. Anal. **32**, 113–139 (2000)

Gallager, R., Humblet, P., Spira, P.: A distributed algorithm for minimum-weight spanning trees. ACM Trans. Programm. Languages Syst. **5**, 66–77 (1983)

Galstyan, A., Krishnamachari, B., Lerman, K., Pattem, S.: Distributed online localization in sensor networks using a moving target. In: IEEE Information Processing in Sensor Networks (IPSN), April 2004, pp. 61–70 (2004)

Galton, A.: Qualitative Spatial Change. Oxford University Press, Oxford (2000)

Gamerman, D., Moreira, A.R.B.: Multivariate spatial regression models. J. Multivariate Anal. **91**, 262–281 (2004)

Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design patterns – Elements of Reusable Object-Oriented Software, 1st edn., Addison Wesley Professional, London (2001)

Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional Computing Series. Addison-Wesley, New York, NY (1995)

Gandin, L.S.: Objective Analysis of Meteorological Fields. Gidrometeorologicheskoe Izdatel'stvo (GIMIZ), Leningrad (1963) (translated by Israel Program for Scientific Translations, Jerusalem, 1965)

Ganesh, A., Massoulie, L., Towsley, D.: The Effect of Network Topology on the Spread of Epidemics. *IEEE Infocom* 2005, Miami, FL (2005)

Gangnon, R.: Impact of prior choice on local Bayes factors for cluster detection. Stat. Med. **25**, 883–895 (2006)

Gao, J. Revesz, P.: Voting prediction using new spatiotemporal interpolation methods. In: Proc. of the Seventh Annual International Conference on Digital Government Research, San Diego (2006)

Garcia-Molina, H., Ullman, J.D., Widom, J.: Database Systems: The Complete Book. Prentice Hall, Upper Saddle River, NJ (2002)

Garcia, Y.J., Lopez, M.A., Leutenegger, S.T.: On optimal node splitting for r-trees. In: Proc. of Very Large Data Bases (VLDB), New York, USA, 24–27 Aug 1998, pp. 334–344

Garg, N., Konjevod, G., Ravi, R.: A Polylogarithmic Approximation Algorithm for the Group Steiner Tree Problem. In: Proc. of SODA (1998)

Garmin, http://www.garmin.com/garmin/cms/site/us (2006)

Garofalakis, M., Kumar, A.: Wavelet synopses for general error metrics. ACM Trans. Database Syst. **30**(4), 888–928 (2005)

Garry, S.: Ohio Land Use Land Cover Classification System. Remote Sensing Program Miscellaneous Report No. 17, Fifth Edition (1988)

Gartner, G. (ed.): Location Based Services & Telecartography, Proceedings of the Symposium 2004, January 28–29, Geowissenschaftliche Mitteilungen. number 66, Technische Universität Wien, Wien (2003)

Garzon, M.B., Blazek, R., Neteler, M., Dios, R.S., Ollero, H.S., Furlanello, C.: Predicting habitat suitability with machine learning models: The potential area of pinus sylvestris l. in the iberian peninsula. Ecol. Modell. **197**(3–4), 383–393 (2006)

Gaunt, C., Jackson, L.: Models for assessing the effects of community change on land use patterns. In: Geertman, S., Stillwell, J. (eds.) Planning Support Systems in Practice, pp. 351–371. Springer, New York, NY (2003)

Gaver, W.: Technology Affordances. Human Factors in Computing Systems, CHI'91 Conference Proceedings, p. 79–84. ACM Press, New York (1991)

Gavoille, C., Peleg, D., Perennes, S., Raz, R.: Distance labeling in graphs. In: Symposium on Discrete Algorithms, pp. 210–219 (2001)

Gavrilova, M.: Weighted voronoi diagrams in biology. http://pages.cpsc.ucalgary.ca/~marina/vpplants/

GDAL: GDAL – Geospatial Data Abstraction Library (2007) http://www.gdal.org. Accessed 22 Feb 2007

Geary, R.C.: The contiguity ratio and statistical mapping. Inc. Stat. **5**, 115–145 (1954)

Gedik, B., Liu, L.: A Customizable *k*-Anonymity Model for Protecting Location Privacy. In: Proceedings of the International Conference on Distributed Computing Systems (ICDCS). pp. 620–629. Columbus, OH, USA (2005)

Gedik, B., Liu, L.: Location privacy in mobile systems: A personalized anonymization model. In: ICDCS, pp. 620–629. IEEE Computer Society (2005)

Gedik, B. Liu, L.: Mobieyes: A distributed location monitoring service using moving location queries. IEEE Transactions on Mobile Computing. **5**(10) (2006)

Gedik B., Liu, L.: Mobieyes: Distributed processing of continuously moving queries on moving objects in a mobile system. In: EDBT, 2004

GE Energy: Smallworld Core Spatial Technology. GE Energy Web Site, http://www.gepower.com/prod_serv/products/gis_software/en/core_spatial.htm (2007)

Geertman, S., Stillwell, J. (eds.): Planning Support Systems in Practice. Springer, New York, NY (2003)

Geffner, S., Agrawal, D., El Abbadi, A.: The Dynamic Data Cube. In: EDBT, pp: 237–253 (2000)

Gelfand, A.E., Ghosh, S.K.: Model Choice: a minimum posterior predictive loss approach. Biometrika **85**, 1–11 (1998)

Gelfand, A.E., Schmidt, A.M., Banerjee, S., Sirmans, C.F.: Nonstationary multivariate process modeling through spatially varying coregionalization (with discussion). TEST **13**, 1–50 (2004)

Gelfand, A.E., Vounatsou, P.: Proper multivariate conditional autoregressive models for spatial data analysis. Biostatistics **4**, 11–25 (2003)

Gelfand, A.E., Zhu, L., Carlin, B.P.: On the change of support problem for spatio-temporal data. Biostat. **2**, 31–45 (2001)

Gellersen, H.-W., Schmidt, A., Beigl, M.: Adding some smartness to devices and everyday things. In: IEEE Workshop on Mobile Computing Systems and Applications, pp. 3–10. December (2000)

Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B.: Bayesian Data Analysis, 2nd edn. Chapman and Hall/CRC Press Boca Raton, FL (2004)

Gelman, A.: Prior distributions for variance parameters in hierarchical models. Bayesian Anal. **3**, 515–533 (2006)

Gendreau, M., Laporte, M.G., Potvin, J.-Y.: Metaheuristics for the capacitated VRP. In: Toth, P., Vigo, D. (eds.) The Vehicle Routing Problem, pp. 129–154. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia (2002)

Geneletti, D.: A GIS-based decision support system to identify nature conservation priorities in an alpine valley. Land Use Policy **21**(2), 149–160 (2004)

GEO (2005) The Global Earth Observation System of Systems (GEOSS) 10-Year Implementation Plan. http://earthobservations.

org/docs/10-Year%20Implementation%20Plan.pdf. Accessed 20 Aug 2007

GEO (2007) About GEO. http://www.earthobservations.org/about/about_GEO.html. Accessed 25 May 2007

Geography Markup Language (GML) Version 3.1.1. http://portal.opengeospatial.org/files/?artifact_id=4700

George, B., Shekhar, S.: Time-aggregated Graphs for Modeling Spatio-temporal Networks – An Extended Abstract. In: Proceedings of Workshops at International Conference on Conceptual Modeling(ER), November (2006)

Georgiadou, Y., Bernard, L., Sahay, S.: Implementation of spatial data infrastructures in transitional economies. Inf. Technol. Dev. **12**, 247–253 (2006)

GEOS (2006) Geometry Engine Open Source. http://geos.refractions.net. Accessed 1 Jan 2007

GEOS (2007): GEOS homepage (2007) http://geos.refractions.net. Accessed 22 Feb 2007

Geospatial Interoperability Reference Model (GIRM, V 1.1) http://gai.fgdc.gov/. Accessed December 2003

Geospatial One-Stop: Creating and Publishing in Support of Geospatial One-Stop and the National Spatial Data Infrastructure. http://www.fgdc.gov/metadata/metadata-publications-list (2004)

Geospatial One Stop (GOS), www.geodata.gov The Geospatial One Stop web portal enables users to search a catalog of geospatial information containing thousands of metadata records, view metadata, download data and services, and interact with Web services

Geospatial One Stop Standards – Digital Ortho Imagery (2002) http://www.geo-one-stop.gov/Standards/DigitalOrthoImagery/index.html. Accessed 11 Oct 2003

Geo VAR-Geospatial Data and Information System http://www.geovar.com/data/satellite/spot/spot.htm?gclid=CNH6OS8ooQCFQ5vOAodVR8rgg Accessed 23 March 2007

Gerke, M., Butenuth, M., Heipke, C., Willrich, F.: Graph-supported verification of road databases. ISPRS J. Photogramm. Remote Sensing **58**, 152–165 (2004)

Germann, T.C., Kadau, K., Longini Jr., I.M., Macken, C.A.: Mitigation strategies for pandemic influenza in the United States. Proc. of National Academy of Sciences (PNAS), vol. 103(15), 5935–5940 (2006)

Gesch, D., Oimoen, M., Greenlee, S., Nelson, C., Steuck, M., Tyler, D.: The national elevation dataset. Photogrammetric Eng. Remote Sensing **68**(1), 5–12 (2002)

Gesink-Law, D.C., Bernstein, K., Serre, M.L., Schumacher, C.M., Leone, P.A., Zenilman, J.M., Miller, W.C., Rompalo, A.M.: Modeling a syphilis outbreak through space and time using the BME approach. Ann. Epidemiol. **16**, 797–804 (2006)

Getis, A, Mur, J., Zoller, H.G. (eds.): Spatial Econometrics and Spatial Statistics. Palgrave MacMillan, New York (2004)

Getis, A., Ord, J.K.: Local spatial statistics: an overview. In: Longley, P., Batty, M. (eds.): Spatial Analysis: Modelling in a GIS Environment, pp. 261–277. Geoinformation International, Cambridge, England (1996)

Geurts, P.: Pattern extraction for time series classification. In: Proceedings of Principles of Data Mining and Knowledge Discovery, 5th European Conference, Freiburg, Germany, 3–5 September 2001, pp. 115–127

Gey, F., Larson, R., Sanderson, M., Joho, H., Clough, P.: GeoCLEF: the CLEF 2005 Cross-Language Geographic Information Retrieval Track. In: The Working Notes for the CLEF 2005 Workshop (2005)

Ghose, R.: Use of information technology for community empowerment: Transforming geographic information systems into community information systems. Trans. GIS **5**, 141–163 (2001)

Ghosh, D. Acharya, R.: A probabilistic scheme for hierarchical routing. In: Proceedings of Ninth IEEE International Conference on Networks, Bangkok, Thailand, 10–12 2001

Ghosh, A., Rushton, G. (eds.): Spatial Analysis and Location-Allocation Models. Van Nostrand Reinhold, New York (1987)

Ghrist, R., Muhammad, A.: Coverage and hole-detection in sensor networks via homology. In: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, Los Angeles, CA, 25–27 Apr 2005

Gibson, G., Nagle, D., Amiri, K., Butler, J., Chang, F.W., Gobioff, H., Hardin, C., Riedel, E., Rochberg, D., Zelenka, J.: File server scaling with network-attached secure disks. In: Proc. of Intl. Conf. on Measurement and Modeling of Computer Systems, SIGMETRICS, Seatle, Washington, June 1997, pp. 272–284

Gibson, J.: The Ecological Approach to Visual Perception. 332. Houghton Mifflin Company, Boston (1979)

Gimblett, H., Itami, R., Durnota, D.: Some Practical Issues in Designing and Calibrating Artificial Human-Recreator Agents in GIS-based Simulated Worlds. Complexity International, 3 (1996)

GIS Standards. http://fortboise.org/maps/GISstandards.html

The GiST Indexing Project (1999) GiST: A Generalized Search Tree for Secondary Storage. http://gist.cs.berkeley.edu. Accessed 1 Jan 2007

Givens, G.H., Hoeting, J.A.: Computational Statistics. Wiley, New Jersey (2005)

Glass, G.: Web Services: Building Blocks for Distributed Systems, Prentice Hall, Upper Saddle River, NJ (2002)

Global Change Master Directory. http://gcmd.nasa.gov/index.html. Accessed November 2006

Globus (2006) Globus Toolkit 4.0 Release Manuals. http://www.globus.org/toolkit/docs/4.0/. Accessed 22 Sept 2006

Glover, E., Mackaness, W.A.: Dynamic generalisation from a single detailed database to support web based interaction. In: Proceedings 19th International Cartographic Conference, Ottawa, 14–21 August (1999)

G.M. Marakas, G.M.: Decision support systems in the 21st centry, 2nd edn. Prentice-Hall, Upper Saddle River (2003)

Gneiting, T., Schlather, M.: Space-time covariance models. In: El-Shaarawi, A.H., Piegorsch, W.W. (eds.) Encyclopedia of Environmetrics, Wiley, Chichester, 2041–2045 (2002)

Goel, S., Imielinski, T., Ozbay, K., Nath, B.: Grassroots: A scalable and robust information architecture, Technical Report DCS-TR-523, Department of Computer Science, Rutgers University, June 2003

Goethals, B., Muhonen, J., Toivonen, H.: Mining non-derivable association rules. SIAM International Conference on Data Mining, 239–249, Newport Beach, California (2005)

Goffman, E.: Dramaturgie des Rollenhandelns. In: Retter, H.: Studienbuch Pädagogische Kommunikation. Klinkhardt, Bad Heilbrunn (2002)

Goldberger, A.: Best Linear Unbiased Prediction in the Linear Model, Journal of the American Statistical Association, vol. 57, pp. 369–375 (1962)

Golden, B.L., Wasil, E.A., Kelly, J.P., Chao, I.-M.: The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results. In: Crainic, T., Laporte, G. (eds.) Fleet management and logistics, pp. 33–56. Kluwer, Boston (1998)

Goldin, D., Kanellakis, P.: On similarity queries for time-series data. In Proc. of CP ’95, September (1995)

Goldin, D., Kutlu, A., Song, M., Yang, F.: The Constraint Database Framework: Lessons Learned from CQA/CDB. In: Proc. International Conference on Data Engineering, pp. 735–737, Bangalore, India, 2003

Goldschlager, L.M.: Short algorithms for space-filling curves. Software–Practice and Experience, SPE **11**(1), 99–100 (1981)

Goldsmith, A.: Wireless Communications. Cambridge University Press (2005)

Goldstein, J., Ramakrishnan, R., Shaft, U., Yu, J.: Processing Queries By Linear Constraints. In: Proceedings of the 16th ACM PODS Symposium on Principles of Database Systems, Tucson, Arizona, USA, 13–15 May 1997, pp. 257–267

Goldstein, K.: The Organism. Aholistic Approachto Biology Derived from Pathological Data in Man. Zone Books, New York (2000, 1934)

Golledge, R.G.: Human wayfinding and cognitive maps. In: Golledge, R.G. (ed.) Wayfinding behavior: Cognitive mapping and other spatial processes, pp. 5–45. Johns Hopkins Press, Baltimore (1999)

Gonzalez, R. C., Woods, R. E.: Digital Image Processing. 2nd edn. Pearson Education, New Delhi, India (2002)

Gooch A., Gooch B.: Non-Photorealistic Rendering. A K Peters Ltd, Wellesley, MA (2001)

Gooch, A., Gooch, B., Shirley, P., Cohen, E.:. A non-photorealistic lighting model for automatic technical illustration. Computer Graphics (Proceedings of SIGGRAPH 1998), pp. 447–452 (1998)

Goodall, C., Mardia, K.V.: Challenges in multivariate spatio-temporal modeling. In: Proceedings of the XVIIth International Biometric Conference, Hamilton, Ontario, Canada, 8–12 August 1994

Goodchild, M., Anselin, L., Deichmann, U.: A Framework for the Aerial Interpolation of Socioeconomic Data. Environ. Planning A **25**, 383–397 (1993)

Goodchild, M.F.: A general framework for error analysis in measurement-based GIS. J. Geogr. Syst. **6**(4), 323–324 (2004)

Goodchild, M.F.: The Geolibrary, Innovations in GIS V. In: Carver, S. (ed.), pp. 59–68. Taylor and Francis, London (1998)

Goodchild, M.F., Guoqing, S., Shiren, Y.: Development and test of an error model for categorical data. Int. J. Geogr. Inf. Syst. **6**, 87–10 (1992)

Goodchild, M.F., Hunter, G.J.: A Simple Positional Accuracy Measure for Linear Features. Int. J. Geogr. Inf. Sci. **11**(3), 299–306 (1997)

Goodchild, M.F., Janelle, D.G.: Spatially Integrated Social Science. Oxford University Press, New York (2004)

Goodchild, M.F., Steyaert, L.T., Parks, B.O. (eds.): GIS and Environmental Modeling: Progress and Research Issues. GIS World Books (1992)

Goodchild, M.F.: Uncertainty: the Achilles heel of GIS? Geo. Info. Syst. November, 50–52 (1998)

Goodchild, M.F., Yuan, M., Cova, T.J.: Towards a general theory of geographic representation in GIS. Int. J.~Geogr. Inf. Sci. **21**, 239–260 (2007)

Goodchild, M.: Geographical Data Modeling. Computers and Geosciences. **18**(4), 401–408 (1992)

Goodchild, M. Gopal, S. (eds.): Accuracy of Spatial Databases. Taylor & Francis, London (1989)

Goodchild, M., Lam, N.: Aerial interpolation: a variant of the traditional spatial problem. Geo-Processing **1**, 297–312 (1980)

Goodday, J.M., Cohn, A.G.: Conceptual neighborhoods in temporal and spatial reasoning. In: ECAI-94 Spatial and Temporal Reasoning Workshop (1994)

Google Earth. http://earth.google.com. Accessed 20 Dec 2006

Google. Google Maps. October 12, (2006). Search on “gas stations” near the “metrodome”

Google Inc. (2006): Google Earth, http://earth.google.com/, accessed 2007-01-05

Google Maps, http://maps.google.com (2007)

Goovaerts, P.: Geostatistical analysis of disease data: accounting for spatial support and population density in the isopleth mapping of cancer mortality risk using area-to-point Poisson kriging. Int. J. Health Geogr. **5**, article 52 (2006)

Goovaerts, P.: Geostatistical analysis of disease data: estimation of cancer mortality risk from empirical frequencies using Poisson kriging. Int. J. Health Geogr. **4**, article 31 (2005)

Goovaerts, P.: Geostatistical analysis of disease data: visualization and propagation of spatial uncertainty in cancer mortality risk using Poisson kriging and p-field simulation. Int. J. Health Geogr. **5**, article 7 (2006)

Goovaerts, P.: Geostatistics for Natural Resources Evaluation. Oxford, New York (1997)

Gordon, M.K.: NGA’s Historian Reflects on First 10 Years, Pathfinder 4(5) Sept.–Oct. National Geospatial Intelligence Agency (2004)

Gottsegen, J., Goodchild, M., Church, R.: A Conceptual Navigable Database Model for Intelligent Vehicle Highway Systems, In: Proc. GIS/LIS, pp. 371–380 (1994)

Gotway, C.A., Young, L.J.: Combining incompatible spatial data. J. Am. Stat. Assoc. **97**, 632–648 (2002)

Gotway, C.A., Young, L.J.: A geostatistical approach to linking geographically-aggregated data from different sources. Technical report #2004–012. Department of Statistics, University of Florida (2004)

Gouraud, H.: Continuous shading of curved surfaces. In Rosalee Wolfe (ed.), Seminal Graphics: Pioneering efforts that shaped the field. ACM Press, New York (1998)

Gow, G.A.: Privacy and ubiquitous network societies. Paper presented at the ITU workshop on ubiquitous network societies, International Telecommunication Union, April 6–8, (2005) Geneva, Switzerland

GPSBabel Community (2007): GPSBabel: convert, upload, download data from GPS and Map programs (2007) http://www.gspbabel.org/. Accessed 20 Feb 2007

Grabow, D.: Overland Flooding Risk Assessment in Rural Communities. Master’s Thesis, University of North Dakota (2004)

GRASS (2006) Homepage of the GRASS project. http://www.grass.itc.it/. Accessed 21 Dec 2006

GRASS Development Team. http://grass.itc.it/. online, 12 (2006)

Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., Pirahesh, H.: Data Cube: A relational aggregation operator generalizing group-by, cross-tab, and sub totals. Data Min. Knowl. Disc. **1**, 29–53 (1997)

Gray, P.: Manager’s guide to making decisions about information systems. John Wiley and Sons, New York (2006)

Gray, P., Watson, H.: Decision support in the data warehouse. Prentice Hall, Upper Saddle River (1998)

Grejner-Brzezinska, D.: Positioning and tracking approaches and technologies. In: Karimi, H.A., Hammad, A. (eds.) Telegeoinformatics: Location-based Computing and Services, pp. 69–110. CRC Press, Boca-Raton, FL (2004)

Griffith, D.: Advanced Spatial Statistics. Kluwer, Dordrecht (1988)

Griffith, D.A., Csillag, F.: Exploring relationships between semi-variogram and spatial autoregressive models. Papers Reg. Sci. **72**(3), 283–295 (1993)

Griffith, D.A.: Some guidelines for specifying the geographic weights matrix contained in spatial statistical models. In: Arlinghous, S.L., Griffith, A.D., Drake, W.D., Nystuen, J.D. (eds.) Practical Handbook of Spatial Statistics, pp. 65–82, CRC, Boca Raton, Florida, USA (1995)

Griffiths, J.: An algorithm for displaying a class of space-filling curves. Software-Practice and Experience **16**(5), 403–411 (1986)

GriPhyN (2006) Grid Physics Network. http://www.griphyn.org/projinfo/intro/appl_domains.php. Accessed 25 Sept 2006

Grohmann, C.H.: Morphometric analysis in geographic infromation systems: applications of free software grass and r. Comput. & Geosci. **30**, 1055–1067 (2004)

Gruber, T.R.: A translation approach to portable ontologies. Knowledge Acquisition **5**, 199–220 (1993)

Gruber, T.: Toward principles for the design of ontologies used for knowledge sharing. Int. J. Hum.-Comput. Stud. **43**, 907–928 (1995)

Gruen, A., Li, H.: Semi-automatic Linear Feature Extraction by Dynamic Programming and LSB-Snake. Photogrammetric Eng. Remote Sensing **63**(8), 985–995 (1997)

Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. In: WWW '004 (2004)

Grumbach, S.: Implementing linear constraint databases. In: Gaede, V., Brodsky, A., Günther, O., Srivastava, D., Vianu, V., Wallace, M. (eds.) Proceedings of the 2nd Workshop on Constraint Databases and Applications, Springer-Verlag, Berlin, New York, 1997

Grumbach, S., Rigaux, P., Scholl, M., Segoufin, L.: DEDALE, a spatial constraint database. In: Cluet, S., Hull, R. (eds.) Proceedings of the 6th International Workshop on Database Programming Languages (DBPL), Lecture Notes in Computer Science, vol. 1369, pp. 38–59. Springer, Berlin (1998)

Grumbach, S., Rigaux, P., Segoufin, L.: The {DEDALE} system for complex spatial queries. In: Proc. ACM SIGMOD International Conference on Management of Data, pp. 213–224, Seattle, Washington, 1998

Grumbach, S., Su, J., Tollu, C.: Linear constraint query languages: expressive power and complexity. In: Leivant, D. (ed.) Logic and Computational Complexity. Lecture Notes in Computer Science, vol. 960, pp. 426–446, Springer-Verlag, Heidelberg (1995)

Grünreich, D.: Development of computer-assisted generalization on the basis of cartographic model theory. In: Müller, J.C., Lagrange, J.P., Weibel, R. (eds.) GIS and Generalization – Methodology and Practice, pp. 47–55. Taylor & Francis, London (1995)

Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceedings of the First International Conference on Mobile Systems, Applications, and Services, San Francisco, CA, pp. 31–42. ACM Press, New York, NY, USA (2003)

Gruteser, M., Hoh, B.: On the anonymity of periodic location samples. In: Hutter, D., Ullmann, M. (eds.) SPC, Lecture Notes in Computer Science, vol. 3450, pp. 179–192. Springer (2005)

Gruteser, M., Liu, X.: Protecting Privacy in Continuous Location-Tracking Applications. IEEE Secur. Priv. **2**(2), 28–34 (2004)

GSDI Association, www.gsdi.org The GSDI Association is an inclusive organization of organizations, agencies, firms, and individuals from around the world. The purpose of the organization is to promote international cooperation and collaboration in support of local, national and international spatial data infrastructure developments that will allow nations to better address social, economic, and environmental issues of pressing importance

Guestrin, C., Bodik, P., Thibaux, R., Paskin, M.A., Madden, S.: Distributed regression: an efficient framework for modeling sensor network data. In: Proceedings of the third international symposium on information processing in sensor networks, Berkeley, California, USA, 26–27 Apr 2004, pp. 1–10. ACM Press (2004)

Guestrin, C., Krause, A., Singh, A.P.: Near-optimal Sensor Placements in Gaussian Processes. In: Proceedings of the Twenty-Second International Conference (ICML 2005), pp. 265–272. (2005)

Gülch E., Müller, H.: New application of semi-automatic building acquisition. In: Baltsavias, E., Grün, A., van Gool, L. (eds.) Automatic Extraction of Man-Made Objects from Aerial and Space Images (III), pp. 103–114. Balkema, Lisse (2001)

Gunther, O., Muller, R.: From GISystems to GIServices: Spatial Computing on the Internet Marketplace. Proceedings of the International Conference on Interoperating Geographic Information Systems, Santa Barbara, CA, December 3–4 1997

Gupta, P., Kumar, P.: The capacity of wireless networks. IEEE Trans. Inf. Theory **46**, 388–404 (2000)

Gupta, S., Kopparty, S., Ravishankar, C.: Roads, codes, and spatiotemporal queries. In: PODS '04: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 115–124, ACM Press, New York, NY, USA (2004)

Guptill, S., Morrison, J.: Elements of spatial data quality. Elsevier, Oxford (1995)

Guralnik, V., Srivastava, J.: Event detection from time series data. In: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, 15–18 August 1999, pp. 33–42

Gurnell, A.M., Montgomery, D.R.: Hydrological Applications of GIS. John Wiley and Sons, New York (2000)

Gustafson, P.: Measurement Error and Misclassification in Statistics and Epidemiology. Chapman & Hall, London (2004)

Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U.: Particle filters for positioning, navigation and tracking. IEEE Trans. Signal Process. **50**, 425–437 (2002)

Gustafsson, F., Gunnarsson, F.: Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements. IEEE Signal Processing Mag., **22**(4), 41–53 (2005)

Güting, R.H., de Almeida, V.T., Ding, Z.: Modeling and querying moving objects in networks. VLDB Journal **15**(2), 165–190 (2006)

Güting, R.H., Böhlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N.A., Nardelli, E., Schneider, M., Viqueira, J.R.R.: Spatio-temporal Models and Languages: An Approach Based on Data Types. In: Sellis, T., et al.(eds.) Spatio-temporal Databases: The CHOROCHRONOS Approach, LNCS 2520, pp. 117–176. Springer, Berlin, Heidelberg, New York (2003)

Güting, R.H., Böhlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N.A., Schneider, M., Vazirgiannis, M.: A Foundation for Representing and Querying Moving Object. In: ACM Transactions on Database Systems, Vol.25, No.1, pp. 1–42. ACM Press, New York, NY, USA (2000)

Güting, R.H., Schneider, M.: Moving Objects Databases. Morgan Kaufmann, San Francisco, CA, USA (2005)

Güting, R.: An introduction to spatial database systems. VLDB J. **3**(4), 357–399 (1994)

Guttman, A.: R-Trees: A Dynamic Index Structure for Spatial Searching. In: Proceedings of the ACM SIGMOD Conference, pp. 47–57. Boston, Massachusetts, USA, 18–21 June 1984

Guttorp, P.: Environmental statistics—a personal view. Int. Stat. Review **71**, 169–179 (2003)

Guttorp, P.: Setting Environmental Standards: A Statistician's Perspective. NRCSE Technical Report 048, http://www.nrcse.washington.edu/pdf/trs48_setting.pdf (2000)

Gutzler, D., Nims, J.: Interannual Variability of water demand and summer climate in Albuquerque, New Mexico. J. Appl. Meteorology. **44**, 1777–1787 (2005)

Guvenc, I., Abdallah, C.T., Jordan, R., Dedeoglu, O.: (2003). Enhancements to RSS based indoor tracking systems using Kalman filtersIn: Global Signal Processing Expo and International Signal Processing Conference, Dallas, Texas, March 31–April 3 2003, Guvenc

Haala, N., Brenner, C.: Virtual City Models from Laser Altimeter and 2D Map Data. Photogramm. Eng. Remote Sensing **65**, 787–795 (1999)

Haas, T.C.: Modeling waterbody eutrophication with a Bayesian belief network. School of Business Administration, Working Paper. University of Wisconsin, Milwaukee, Wisconsin (1998)

Hadjieleftheriou, M., Hoel, E., Tsotras, V.J.: Sail: A library for efficient application integration of spatial indices. In: Proc. of Scientific and Statistical Database Management (SSDBM), Santorini Island, Greece, 21–23 2004, pp. 135–138

Hadjieleftheriou, M., Hoel, E., Tsotras, V.J.: Sail: A spatial index library for efficient application integration. GeoInformatica **9**(4):367–389 (2005)

Hadjieleftheriou, M., Kollios, G., Bakalov, P., Tsotras, V.J.: Complex Spatio-temporal Pattern Queries. In: Proceedings of the 31st International Conference on Very Large Data Bases (VLDB'05), pp. 877–888 (2005)

Hadjieleftheriou, M., Kollios, G., Bakalov, P., Tsotras, V.J.: Complex spatio-temporal pattern queries. In: VLDB 05: Proceedings of the 31st international conference on Very large data bases, pp. 877–888, Trondheim, Norway, 30 Aug–2 Sept 2005

Hadjieleftheriou, M., Kollios, G., Tsotras, V.J., Gunopulos, D.: Efficient indexing of spatiotemporal objects. In: Proc. of Extending Database Technology (EDBT), Prague, 24–28 Mar 2002, pp. 251–268

Hadjieleftheriou, M., Kollios, G., Tsotras, V.J., Gunopulos, D.: Indexing spatiotemporal archives. VLDB J. **15**(2), 143–164 (2006)

Hadjieleftheriou, M.: SaIL. http://spatialindexlib.sourceforge.net

Hadzilacos. T.H., Tryfona, N.: Extending the Entity-Relationship Model to Capture Spatial Semantics., SIGMOD RECORD. 26 (3), 24"29 (1997)

Haeberlen, A., Flannery, E., Ladd, A.M., Rudys, A., Wallach, D.S., Kavraki, L.E.: Practical robust localization over large-scale 802.11 wireless networks: In: Proceedings of the Tenth ACM Int. Conf. on Mobile Computing and Networking (MOBICOM), Philadelphia, PA, pp. 70–84 (2004)

Haesevoets, S.: Modelling and Querying Spatio-temporal Data. PhD thesis, Limburgs Universitair Centrum (2005)

Hagani, A., Oh, S.C.: Formulation and Solution of a Multi-Commodity, Multi-Modal Network Flow Model for Disaster Relief Operations. Transp. Res. Part A **30**, 231–250 (1996)

Hage, C., Jensen, C.S., Pedersen, T.B., Speicys, L., Timko, I.: Integrated Data Management for Mobile Services in the Real World, In: Proc. International Conference on Very Large Data Bases, pp. 1019–1030 (2003)

Hagedoorn, M.: Nearest Neighbors Can Be Found Efficiently If the Dimension Is Small Relative to the Input Size. In: Proceedings of the 9th International Conference on Database Theory - ICDT 2003. volume 2572 of Lecture Notes in Computer Science, pp. 440–454, Springer, January (2003)

Hägerstrand, T.: What about people in regional science. Pap. Region. Sci. Assoc. **24**, 7–21 (1970)

Haggett, P., Chorley, R.J.: Network Analysis in Geography. Edward Arnold, London (1969)

Haggett, P.: Locational Analysis in Human Geography. St Martin's Press, New York (1966)

Haining, R.P.: Spatial Data Analysis: Theory and Practice. Cambridge University Press, Cambridge (2003)

Haining, R.: Spatial Data Analysis in the Social and Environmental Sciences. Cambridge University Press, Cambridge, UK (1993)

Halevy, A.Y., Ives, Z.G., Suciu, D., Tatarinov, I.: Piazza: Data management infrastructure for semantic web applications. In: Proceedings of the International Conference on Data Engineering (ICDE) 2003, March 5–8, Bangalore, India

Hall, D.L.: Mathematical Techniques in Multisensor Data Fusion. Artech, Boston, MA (1992)

Halloran, M.E.: Concepts in infectious disease epidemiology. In: Rothman, K.J., Greenland, S. (eds.) Modern Epidemiology, 2nd edn. Lippincott-Raven, Philadelphia (1998)

Hall, P., Dowling, G.: Approximate String Matching. ACM Computing Surveys **12**(4), 381–402 (1980)

Hamming, R.W.: Error-detecting and error-correcting codes. In: Bell System Technical Journal, pp. 29(2):147–160. Bell System (1950)

Hampe, M., Sester, M.: Generating and using a multi-representation database (MRDB) for mobile applications. Paper presented at the ICA Workshop on Generalization and Multiple Representation, Leicester, 20–21 August 2004. Available via: http://ica.ign.fr/Leicester/paper/hampe-v2-ICAWorkshop.pdf. Accessed 12 Feb 2007

Hampe, M., Sester, M., Harrie, L.: Multiple representation databases to support visualisation on mobile devices. In: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 35, ISPRS, Istanbul (2004)

Handcock, M.S., Stein, M.L.: A Bayesian analysis of kriging. Technometrics. **35**, 403–410 (1993)

Hand, D. J., Mannila, H., Smyth, P.: Principles of Data Mining. MIT Press, Cambridge, MA (2001)

Haneuse, S., Wakefield, J.: Ecological inference incorporating spatial dependence. In: King, G., Rosen, O., Tanner, M. (eds.) Ecological Inference: New Methodological Strategies, chapter 12, pp 266–301. Cambridge University Press, London (2004)

Hanewinkel, C., Tzschaschel, S.: Germany in maps – a multipurpose toolbox In: Proceedings of the 22th International Cartographic Conference A Coruña, 9–16 July 2005: CD-ROM

Han, A.F.: TEVACS: Decision Support System for Evacuation Planning in Taiwan. J. Transp. Eng. **116**, 821–830 (1990)

Han, J., Kamber, M.: Data Mining: Concepts and Techniques (2nd ed.). Morgan Kaufmann, London, UK (2006)

Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, San Francisco, CA (2001)

Han, J., Stefanovic, N., Koperski, K.: Selective materialization: An efficient method for spatial data cube construction. In: Second Pacific-Asia Conference, PAKDD-98, pp. 144–158, Melbourne, Australia, 15–17 Apr 1998

Han, J., Wang, J., Lu, Y., Tzvetkov, P.: Mining top-k frequent closed patterns without minimum support. In: Proc. of Intl. Conf. on Data Mining, pp. 211–218. 9–12 Dec 2002

Han, L.D., Franzese, O.: Modeling Mass-Evacuation Traffic Operations Due to Hurricanes. Presented at ITE 2001 Annual Meeting and Exhibit, Chicago, Illinois, 2001

Hannaford, C.: Smart Moves: Why Learning is not all in your Head. Great Ocean Publishers, Arlington, VA (1995)

Haralick, R.M.: A spatial data structure for geographic information systems. In: Freeman, H., Pieroni, G.G. (eds.) Map Data Processing, Academic, New York (1980)

Harary, F.: Graph Theory, pp. 1–2. Addison-Wesley, Reading (1994)

Haraway. D.: Situated Knowledges: The Science Question in Feminism and the Priviledge of Partial Perspective. In: Simians, Cyborgs, and Women. Routledge, New York, (1991)

Harinarayan, V., Rajaraman, A., Ullman, J.D.: Implementing data cubes efficiently. In: Proceedings of ACM-SIGMOD International Conference on Management of Data, Montreal, Quebec, Canada, 4–6 June 1996, pp. 205–216

Harrie, L., Sarjakoski, L.T., Lehto, L.: A Mapping Function for Variable-Scale Maps in Small-Display Cartography. J. Geospa. Eng. **4**(2), 111–123 (2002)

Harrie, L, Stigmar, H., Koivula, T., Lehto, L.: An algorithm for icon labelling on a real-time map. In: Fisher, P. (ed.) Developments in Spatial Data Handling, pp. 493–507. Springer-Verlag, Berlin, Heidelberg, New York (2004)

Harries, K.D.: Crime and the Environment. Charles C Thomas Press, Spingfield, Il (1980)

Harries, K.D.: The Geography of Crime and Justice. McGraw-Hill, New York, NY (1974)

Harries, K.: Mapping Crime: Principle and Practice. NCJ 178919, National Institute of Justice, US Department of Justice: Washington, DC. Available at http://www.ncjrs.org/html/nij/mapping/pdf.html (1999)

Harrington, B., Huang, Y.: In-network surface simplification for sensor fields. In: Proceedings of the 13th annual ACM international workshop on geographic information systems, pp. 41–50, Bremen, Germany, 4–5 Nov 2005

Harroud, H., Ahmed, M., Karmouch, A.: Policy-driven personalized multimedia services for mobile users. IEEE Trans. on Mobile Computing **2**(1) 16–24 (2003)

Harrower, M.: A look at the history and future of animated maps. Cartographica **39**, 33–42 (2004)

Hartigan, J.A.: Clustering algorithms, Wiley, New York, xiii, 351 p (1975)

Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, New York (2003)

Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)

Hartsfield, R.: Pearls in Graph Theory. Academic Press, San Diego (1990)

Hartshorne, R.: The Nature of Geography: A Critical Survey of current thought in the light of the past. PA: Association of American Geographers, Lancaster (1939)

Harvey, F., Kuhn, W., Pundt, H., Bishr, Y.: Semantic interoperability: A central issue for sharing geographic information. Ann. Reg. Sci. **33**, 213–232 (1999)

Harvey, H., Han, D.: The relevance of open source to hydroinformatics. J. Hydroinformatics **4**, 219–234 (2002)

Harville, D.: Matrix Algebra from a Statistician's Perspective. Springer-Verlag, New York (1997)

Haslett, J., Bradley, R., Craig, P., Unwin, A., Wills, G.: Dynamic graphics for exploring spatial data with application to locating global and local anomalies. Am. Statistician **45**, 234–242 (1991)

Hassan Ghassemain. On-board satellite image compressiaon by object-feature wxtraction. International Atchieves of Photogrammertry Remote Sensing and Spatial Information Sciences.**35**, 820–825 (2004)

Hatami, A., Pahlavan, K.: Comparative statistical analysis of indoor positioning using empirical data and indoor radio channel models. In: Proceedings of the Consumer Communications and Networking Conference, Las Vegas, Nevada, USA, 8–10 Jan 2006

Haunert, J.H., Sester, M.: Area collapse and road centerlines based on straight skeletons. Geoinformatica (2007, in press)

Haunert, J.H., Sester, M.: Propagating updates between linked datasets of different scales. In: Proceedings of 22nd International Cartographic Conference, pp. 9–16. La Coruna, Spain (2005)

Haunert, J.H., Wolff, A.: Generalization of land cover maps by mixed integer programming. In: Proceedings of 14th International Symposium on Advances in Geographic Information Systems, Arlington, Virginia, USA (2006)

Hautamaki, V., Karkkainen, I., et al.: Outlier Detection Using k-Nearest Neighbor Graph. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004. vol. 3. Aug. 23–26, pp. 430–433, (2004)

Havercroft. M.: Consultation paper: Positional accuracy of large-scale data and products. Ordnance Survey, Southampton (1997)

Haykin S.: Neural Networks a Comprehensive Foundation. Prentice Hall Inc., New York, NY (1999)

Healey, R., Dowers, S., Gittings, B., Mineter, M.: Parallel Processing Algorithms for GIS. Taylor & Francis, Bristol, USA (1998)

Hearnshaw, H.M., Unwin, D.J., (eds.): Visualization in Geographical Information Systems. Wiley, London (1994)

Heckbert, P.S., Garland, M.: Survey of Polygonal Surface Simplification Algorithms. In: SIGGRAPH 97 Course Notes, No. 25, ACM Press, New York (1997)

Heckerman, D.: Bayesian networks for data mining. Data Mining and Knowledge Discovery, **1**, 79–119. MapWindow Open Source Team (2007). MapWindow GIS 4.3 Open Source Software, Accessed February 06, 2007 at the MapWindow Website: http://www.mapwindow.org/ (1997)

Heinemann, A., Kangasharju, J., Lyardet, F., Mühlhäuser, M.: iClouds – P2P information sharing in mobile environments, International Conference on Parallel and Distributed Computing (Euro-Par 2003), Klagenfurt, Austria (2003)

Heinzelman, W.R., Chandrakassan, A.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. Paper presented at the 33rd Hawaii International Conference of System Sciences, Maui, Hawaii, January 2000

Hellerstein, J.M., Haas, P.J., Wang, H.J.: Online aggregation. In: SIGMOD '97: Proceedings of the 1997 ACM SIGMOD International conference on Management of data, pp. 171–182. ACM Press, New York, NY, USA (1997)

Hellerstein, J.M., Hong, W., Madden, S., Stanek, K.: Beyond average: Towards Sophisticated Sensing with Queries. In: Proceedings of the Second International Workshop on Information Processing in Sensor Networks, pp. 63–79, Palo Alto, CA, USA, 22–23 Apr 2003

Hellerstein, J.M., Naughton, J.F., Pfeffer, A.: Generalized search trees for database systems. In: Proceedings of the 21st International Conference on Very Large Data Bases, Zürich, Switzerland, 11–15 Sept 1995

Hendler, J., Berners-Lee, T., Miller, E.: Integrating applications on the semantic web (English version). J. Inst. Electrical Eng. Jpn. **122**, 676–680 (2002)

Hendriks, P., Vriens, D.: From geographical information systems to spatial group decision support systems: a complex itinerary. Geogr. Env. Mod. **4**(1), 83–104 (2000)

Henk, R.H., Ballard, A.J., Robideau, R.L., Peacock, W.G., Maghelal, P., Lindell, M.K., Prater, C.S., Loftus-Otway, L., Siegesmund, P., Harrison, R., Lasdon, L., Chiu, Y.C., Zheng, H., Perkins, J., Lewis, C., Boxill, S.: Disaster Preparedness in Texas. College Station, Texas Transportation Institute (2007)

Hernández, D., Clementini, E., Felice, P.D.: Qualitative distances. In: Frank, A.U., Kuhn, W. (eds.) Proc. COSIT 95, Lecture Notes in Computer Science, vol. 988, pp. 45–57. Springer, Berlin (1995)

Hernandez, M., Miller, S., Goodrich, D.C., Goff, B.F., Kepner, W.G., Edmonds, C.M., Jones, K.B.: Modeling runoff response to land cover and rainfall spatial variability in semi-arid watersheds, Environ. Monit. and Assess. **64**:285–298 (2000)

Herring, J.R.(eds.): OpenGIS Implementation Specification for Geographic Information – Simple Feature Access. http://www.opengeospatial.org/standards/sfs (2006). Accessed 30 Aug 2007

Hershberger, J., Shrivastava, N., Suri, S.: Cluster hulls: a technique for summarizing spatial data streams. In: Proceedings of IEEE International Conference on Data Engineering, Atlanta, Georgia, 2006, p. 138

Herzog, A.: Developing Cartographic Applets for the Internet. In: Peterson, M. (ed.) Maps and the Internet, pp. 117–130. Elsevier, Kidlington, Oxford, UK (2003)

Hess, R.L., Rubin, R.S., West, L.A.: Geographic information systems as a marketing information system technology. Dec. Supp. Sys. **38**(2), 197–212 (2004)

Hethcote, H.: The Mathematics of Infectious Diseases. SIAM Rev. **42**(4), 599–653 (2000)

Hettwer, J., Benning, W.: Restklaffenverteilung mit der Natural-Neighbour-Interpolation. Allg. Vermess.-Nachr. **110**, 122–129 (2003) (in German)

Heuvelink, G.B.M., Burrough, P.A. (eds.) Developments in Statistical Approaches to Spatial Uncertainty and its Propagation. Themed issue Int. J. Geogr. Inf. Sci. **16**, number 2 (2002)

Heuvelink, G.B.M.: Error Propagation in Environmental Modelling with GIS. Taylor & Francis, London (1998)

Heuvelink, G.B.M.: Handling spatial uncertainty in GIS: development of the data uncertainty engine. Instituto Geografico Portugues, Estoril (Portugal) (2005)

Hier-Majumder, C.A., Travis, B.J., Belanger, E., Richard, G., Vincent, A.P., Yuen, D.: A.: Efficient sensitivity analysis for flow and transport in the Earth's crust and mantle. Geophys. J. Int. **166**, 907–922 (2006)

Hightower, J., Borriello, G.: Location systems for ubiquitous computing. IEEE Comput. **34**, 57–66 (2001)

Hightower, J., Borriello, G.: A survey and taxonomy of location systems for ubiquitous computing. In: Technical Report UW CSE 01–08–03, University of Washington, Washington (2001)

Hilbert, D.: Über die stetige Abbildung einer Linie auf ein Flächenstück. Mathematische Annalen, , 459–460 (1891)

Hilborn, R., Mangel, M.: The Ecological Detective: Confronting Models with Data. Princeton University Press, Princeton (1997)

Hill, B.: Catching the Bad Guy. In: Levine, N. (ed.) CrimeStat III: A Spatial Statistics Program for the Analysis of Crime Incident Locations (version 3.0), Chapter 10. Ned Levine & Associates and the National Institute of Justice, Houston, TX, Washington, DC (2004)

Hinz, A., Dörstel, C., Heier, H.: Digital Modular Camera: System concept and data processing workflow. IAPRS, vol. XXXIII, part B1, Amsterdam, pp. 164–171, digitally on CD-Rom. (2000)

Hirtle, S.C., Sorrows, M.E.: Navigation in electronic environments. In: Allen, G. (ed.) Applied Spatial Cognition, pp. 103–126. Lawrence Erlbaum Associates, Mahwah NJ (2007)

Hirtzler, M.: Pilat Informative Educative: SVG – PHP – MySQL – JavaScript (2000–2006), http://pilat.free.fr/english/index.htm, accessed 2006–11–12

Hjaltason, G.R., Samet, H.: Distance browsing in spatial data-bases. ACM Transaction on Database. Syst. (TODS) **24**(2), 265–318 (1999)

Hjaltason, G.R., Samet, H., Sussmann, Y.J.: Speeding up Bulk-Loading of Quadtrees. In: Proceedings of the 5th ACM international workshop on Advances in Geographic Information Systems, University of Maryland, Las Vegas, Nevada, USA (1997)

Hjelm, J.: Creating Location Services for the Wireless Guide, Professional Developer's Guide Series. Wiley Computer Publishing, New York (2002)

Hjorth, U.: Computer Intensive Statistical Methods, Validation, Model Selection, and Bootstrap. Chapman and Hall/CRC, London, New York (1994)

H-L Yu, Christakos, G.: Spatiotemporal modelling and mapping of the bubonic plague epidemic in India. Int.. J. Health Geogr. **5** (2006) Available via: http://www.ij-healthgeographics.com/content/5/1/12

H-L Yu, Kolovos, A., Christakos, G., J-C Chen, Warmerdam, S., B. Dev. Interactive spatiotemporal modelling of health systems: The SEKS-GUI framework. In: Griffith, D.A., Christakos, G. (eds.) J. SERRA–Special Issue on Medical Geography as a Science of Interdisciplinary Knowledge Synthesis under Conditions of Uncertainty (2007, in press)

Hobbs, J.: Granularity. In: Proceedings of the IJCAI 85 (1985)

Hobeika, A.G., Ardekani, S.A., Martinez-Marquez, A.: A Decision Support System for Emergency Transportation Operations Following Major Urban Earthquake: a Case Study of the 1985 Mexico City Earthquake. National Science Foundation ECE-8610841 (1987)

Hobeika, A.G., Jamei, B.: MASSVAC: a Model for Calculating Evacuation Times under Natural Disasters. Presented at Computer Simulation in Emergency Planning. Society for Computer Simulation, San Diego (1985)

Hobeika, A.G., Radwan, A.E., Jamei, B.: Transportation Actions to Reduce Evacuation Times Under Hurricane/Flood Conditions: A Case Study of Virginia Beach City. Presented at Annual Meeting of Transportation Research Board, Washington, DC Jan 1985

Hochmair, H., Lüttich, K.: An Analysis of the Navigation Metaphor – and Why it Works for the World Wide Web. Spat. Cogn. Comp. **6**(2), 235–278 (2006)

Hochschild, J.: Where You Stand Depends on What You See: Connections among Values, Perceptions of Fact, and Prescriptions. In: Kuklinski, J. (ed.) Citizens and Politics: Perspectives from Political Psychology. Cambridge University Press, Cambridge (2000)

Hoekman, D.H., Varekamp, C.: Observation of tropical rain forest trees by airborne high-resolution interferometric radar. Geosci. Remote Sensing, IEEE Trans on, **39**(3), 584–594 (2001)

Hoel, E.G., Heng, W.L., Honeycutt, D.: High Performance Multimodal Networks. In: Proceedings of International Symposium on Spatial and Temporal Databases SSTD (2005)

Hoel, E., Menon, S., Moorhouse, S.: Building a Robust Implementation of Topology. In Advances in Spatial and Temporal Databases. Proceedings of the 8th International Symposium on Spatial and Temporal Databases. SSTD 2003. Santorini, Greece (2003)

Höfle, B., Rutzinger, M., Geist, T., Stötter, J.: Using airborne laser scanning data in urban data management – set up of a flexible information system with open source components. In: Proceedings UDMS 2006: Urban Data Management Symposium, Aalborg, Denmark (2006)

Hoh, B., Gruteser, M.: Protecting location privacy through path confusion. In: Proceedings of IEEE/Create-Net SecureComm 2005, Athens, Greece, September 2005, pp. 194–205. IEEE Computer Society, Washington, DC, USA (2005)

Hoh, B., Gruteser, M., Xiong, H., Alrabady, A.: Enhancing privacy preservation of anonymous location sampling techniques in traffic monitoring systems. In: Proceedings (Poster Session) of IEEE/Create-Net SecureComm 2006, Baltimore, Maryland August 2006, pp. 1–3. IEEE Computer Society, Washington, DC, USA (2006)

Hoh, B., Gruteser, M., Xiong, H., Alrabady, A.: Enhancing security and privacy in traffic-monitoring systems. IEEE Pervasive Comput. **5**(4), 38–46 (2006)

Hoh, B., Gruteser, M., Xiong, H., Alrabady, A.: Preserving Privacy in GPS Traces via Uncertainty-Aware Path Cloaking. In: CCS '07: Proceedings of the 14th ACM conference on Computer and communications security, Alexandria, Virginia, USA. ACM Press, New York, NY, USA (2007, in press)

Holland, G.: An analytic model of the wind and pressure profiles in hurricanes. Monthly Weather Review **108**, 1212–1218 (1980)

Homer C., Huang C., Yang L., Wylie, B., Coan, M.: Development of a 2001 National Land-cover Database for the United States. Photogramm. Eng. Remote Sens. **70**, 829–840 (2004)

Honkela T: (1997) Self-organizing map in natural language processing. http://citeseer.ist.psu.edu/123560.html. Accessed 30 Dec 2006

Hoppe, H.: Progressive meshes. In: SIGGRAPH'96 Proceedings, 4–9 Aug 1996, pp. 99–108. New Orleans, Louisiana, USA (1996)

Hornsby, K., Egenhofer, M., Hayes, P.: Modeling cyclic change. In: Advances in Conceptual Modeling, Proceedings of the ER'99 Workshops, Paris, France., Lecture Notes in Computer Science, vol. 1727, pp. 98–109. Springer-Verlag, Berlin (1999)

Hornsby, K., Egenhofer, M. J.: Modeling moving objects over multiple granularities. Ann. Math. Artif. Intell. **36**, 177–194 (2002)

Hornsby, K., Worboys, M. (eds.): Event-oriented Approaches in Geographic Information Science. Lawrence Erlbaum Associates. Special issue of: Spatial Cognition and Computation, vol. 4, number 1 (2004)

Hosobe, H.: A modular geometric constraint solver for user interface applications. In: Proc. UIST '01, pp. 91–100. ACM (2001)

Hossain, M., Lawson, A.B.: Cluster detection diagnostics for small area health data: with reference to evaluation of local likelihood models. Stat. Med. **25**, 771–786 (2006)

Houser, S.A.: Medical Surface Transformations for Rapid Approximation of Casting Solidification. MS in mechanical engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia (1996)

Houston, S.H., Shaffer, W.A., Powell, M.D., Chen, J.: Comparisons of HRD and SLOSH surface wind fields in hurricanes: Implications for storm surge modeling. Weather and Forecasting **14**, 671–686 (1999)

Hsu, S.A., Yan, Z.: A note on the radius of maximum wind for hurricanes. J. Coastal Res. **14**, 667–668 (1998)

http://www.en.wikipedia.org: This collaborative web-based, free content encyclopedia project is a good resource to see information on WFS brought together. It provides additional links for further investigation

http://www.skyhookwireless.com/

http://www.w3.org/XML/

Huang C., Wylie B., Homer C., Yang L., Zylstra, G.: Derivation of a Tasseled Cap Transformation Based on Landsat 7 At-Satellite Reflectance. Int. J. Remote Sens. **8**, 1741–1748 (2002)

Huang, P.W., Lin, P.L., Lin, H.Y.: Optimizing storage utilization in r-tree dynamic index structure for spatial databases. J. Syst. Softw. **55**(3):291–299 (2001)

Huang, X., Jensen, C. S., Saltenis, S.: The Islands Approach to Nearest Neighbor Querying in Spatial Networks. SSTD (2005)

Huang, Y., Dimitrova, V., Agarwal, P.: Detecting mismatches between a user's and a system's conceptualizations to empower user modelling for the Semantic Web. In: Proceedings of the Workshop on Personalization in the Semantic Web, User Modelling Conference 2005, Edinburgh, 24–30 July 2005

Huang, Y., Jing, N., Rundensteiner, E.: Spatial Joins Using R-trees: Breadth-First Traversal with Global Optimizations. In: VLDB, Athens, Greece, pp. 396–405 (1997)

Huang, Y., Shekhar, S., Xiong, H.: Discovering Co-location Patterns from Spatial Datasets: A General Approach. IEEE TKDE, **16**(12) 1472–1485 (2004)

Huang, Y., Xiong, H., Shekhar, S., Pei, J.: Mining confident co-location rules without a support threshold. In: Proc. 2003 ACM Symposium on Applied Computing (ACM SAC March, 2003), pp. 497–501. ACM Press, Melbourne, Florida (2003)

Huang, Y., Zhang, L., Zhang, P.: Finding sequential patterns from a massive number of spatio-temporal events. In: Proc. of SIAM International Conference on Data Mining (SDM) (2006)

Hubel, D., Wiesel, T.: Receptive fields, binocular interaction & functional architecture in the cat's visual cortex. Journal of Physiology (London) **160**, 106–154 (1962)

Hudson, P.J., Rizzoli, A., Grenfell, B.T., Heesterbeek, H., Dobson, A.P. (eds.): The Ecology of Wildlife Diseases. Oxford University Press, Oxford (2001)

Hue, C., Le Cadre, J.-P., Perez, P.: Posterior Cramer-Rao bounds for multi-target tracking. IEEE Trans. Aerospace Electron. Syst. **42**(1), 37–49 (2006)

Huertas, A., Lin, C., C., Nevatia, R.: Detection of Building form Monocular View of Aerial Scenes using Perceptual Grouping and Shadows. In: Proceeding DARPA Image Understanding Workshop, Washington, DC, pp. 253–260 (1993)

Hughes Applied Information Systems (1994) Proposed ECS Core Metadata Standard, Release 2.0, 420-TP-001-005: Landover, Maryland

Hu, H., and Lee, D.L.: Range nearest-neighbor query. IEEE Trans. on Knowl. Data Eng. **18**(1), 78–91 (2006)

Hu, H., Lee, D.L., Xu, J. Fast Nearest Neighbor Search on Road Networks. EDBT (2006)

Hu., H., Xu, J., Wang, W.S., Zheng, B., Lee, D.L., Lee, W.: Proactive Caching for Spatial Queries in Mobile Environments. Proceedings of the 21st International Conference on Data Engineering (ICDE 2005), Tokyo, Japan, April 2005

Hu, L., Evans, D.: Localization for mobile sensor networks. In: MobiCom '04: Proc. 10th Int. Conf. on Mobile computing and networking, pp. 45–57 (2004)

Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: Proceedings of ACM Special Interest Group on Knowledge Discovery and Data Mining, San Francisco, 2001, pp. 97–106

Humblet JP (ed) (2003) Web Map Context Documents. OpenGIS Project Document 03-036r2. https://portal.opengeospatial.org/files/?artifact_id=3841

Hunter, G.J., Goodchild, M.F.: Communicating uncertainty in spatial databases. Transactions in GIS **1**(1), 13–24 (1996)

Hurni, L.: Interaktive Karteninformationssysteme – quo vaditis? Kartogr. Nachr. **56**, 136–142 (2006)

Idaho Geospatial Data Clearinghouse, Interactive Numeric & Spatial Information Data Engine. http://inside.uidaho.edu/whatsnew/whatsnew.htm (2006)

IGBP – International Geosphere-Biosphere Programme. http://www.igbp.net/. Accessed November 2006

IHDP – International Human Dimensions Programme on Global Environmental Change. http://www.ihdp.org/. Accessed November 2006

Ihekwaba, A.E.C., Broomhead, D.S., Grimley, R.L., Benson, N., Kell, D.B.: Sensitivity analysis of parameters controlling oscillatory signaling in the NF-kB pathway: the roles of IKK and IkBa. Syst. Biol. **1**, 93–103 (2004)

Ihler, A.T., Fisher III, J.W., Moses, R.L.: Nonparametric belief propagation for self-calibration in sensor networks. In: IEEE ICASSP 2004, vol. 3, pp. 861–864 May (2004)

Ilarri, S., Mena, E., Illarramendi, A.: Location-dependent queries in mobile contexts: Distributed processing using mobile agents. ACM TMC, **5**(8), August 2006

Ilg, D., Wynne, D., Fortner, B. (1995) The HDF-EOS Swath Concept. Hughes Applied Information Systems. doi:170-WP-003-001

Imhoff, E.: Positioning names on maps. The American Cartographer **2**(2), 128–144 (1975)

Indulska, J., McFadden, T., Kind, M., Henricksen, K.: Distributed Applications and Interoperable Systems, In: Scalable Location Management for Context-Aware Systems, Lecture Notes in Computer Science, vol. 2893, pp. 224–235. Springer (2003)

Indyk, P., Koudas, N., Muthukrishnan, S.: Identifying representative trends in massive time series data sets using sketches. In: Proceedings of the 26th International Conference on Very Large Data Bases, Cairo, Egypt, 10–14 September 2000, pp. 363–372

Indyk, P., Motwani, R.: Approximate nearest neighbor: towards removing the curse of dimensionality. In: Proceedings of the Symposium on Theory of Computing, pp. 604–613 (1998)

Information technology – digital compression and coding of continuous-tone still images – Part 1: Requirements and guidelines. ISO/IEC International Standard 10918-1, ITU-T Rec. T.81 (1993)

Inkscape Community: Inkscape, SVG Authoring Tool (2000–2006), http://www.inkscape.org, accessed 2006–11–12

Inmon, W.H.: Building the data warehouse. John Wiley and Sons, New York (1990)

Institute of Electrical and Electronics Engineers: IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. IEEE Computer Society, USA (1990)

Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J., Silva, F.: Directed Diffusion for Wireless Sensor Networking. ACM/IEEE Trans. Netw. **11**(1), 2–16 (2002)

Interagency Team: Stream Restoration Guidance Manual. Stream Corridor Restoration: Principles, Processes, and Practices. By the Federal Interagency Stream Restoration Working Group (FISRWG) (15 Federal agencies of the US government). GPO Item No. 0120-A; SuDocs (1998) ISBN-0-934213-59-3

International Journal of Spatial Data Infrastructures Research. ISSN 1725-0463. http://ijsdir.jrc.it/

International Organization for Standardization. http://www.isotc211.org/ (2007)

International Organization for Standardization: ISO 19113 Geographic information – Quality principles. Geneva (2002)

International Spatial Accuracy Research Association. http://www.spatial-accuracy.org

International Standards Organization (1994) Technical Committee 211: GeoInformation, Geomatics. http://www.isotc211.org/

Introduction to GIS–Raster-Based GIS. http://www.sli.unimelb.edu.au/gisweb/GISModule/GIST_Raster.htm Accessed 21 February 2007

Ioannidis, Y.E., Poosala, V.: Histogram-based approximation of set-valued query-answers. In: VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases, pp. 174–185. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1999)

Isenberg, T., Freudenberg, B., Halper, N., Schlechtweg, S., Strothotte T.: A developer's guide to silhouette algorithms for polygonal models. IEEE Computer Graphics and Applications, 23(4):28–37, (2003)

Isenberg, T., Halper, N., Strothotte, T.: Stylizing silhouettes at interactive rates: from silhouette edges to silhouette strokes, In: Computer Graphics Forum (Proceedings of Eurographics), 21(3):249–258 (2002)

Ishikawa, T., Montello, D.R.: Spatial knowledge acquisition from direct experience in the environment: individual differences in the development of metric knowledge and the integration of separately learned places. Cognit. Psychol. **52**, 93–129 (2006)

Iskander, D.R., Collins, M.J., Davis, B., Franklin, R.: Corneal surface characterization: How many zernike terms should be used?(ARVO) abstract. Invest. Ophthalmol. Vis. Sci. **42**(4), 896 (2001)

Iskander, D.R., Collins, M.J., Davis, B.: Optimal modeling of corneal surfaces with zernike polynomials. IEEE Trans Biomed. Eng. **48**(1), 87–95 (2001)

Islam, A.S., Bermudez, L., Beran, B., Fellah, S., Piasecki, M.: Ontology for Geographic Information – Metadata (ISO 19115:2003). http://loki.cae.drexel.edu/~wbs/ontology/iso-19115.htm. Accessed 21 March 2007

Isli, A.: On deciding consistency for CSPs of cyclic time intervals. In: Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference, Key West, FL, 21–23 May 2001

ISO 19115:2003. Geographic Information-metadata. http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=26020. Accessed 29 Apr 12007

ISO 19115-2. Geographic information – Metadata – Part 2: Extensions for imagery and gridded data. http://www.isotc211.org/pow.htm

ISO 19119, Geographic Information—Services (2005)

ISO 19124 Review Summary 2001-01. Geographic Information-Imagery and Gridded Data Components. http://www.isotc211.org/protdoc/211n1017/

ISO 19125:2004. Geographic Information-Simple Feature Access. http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=40114

ISO 19128:2005. Geographic information – Web map server interface. http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=40114

ISO 19129. Geographic information – Imagery, gridded and coverage data framework. http://www.isotc211.org/pow.htm

ISO 19130. Geographic information – Sensor and data models for imagery and gridded data. http://www.isotc211.org/pow.htm

ISO 19136. Geographic Information-Geographic Makeup Language. http://www.isotc211.org/pow.htm

ISO/IEC: ISO/IEC 13249-3: Information Technology—Database Languages—SQL Multimedia and Application Packages. Part 3: Spatial (2006)

ISO/IEC: ISO/IEC, ISO/IEC TR 14252 Information technology – Guide to the POSIX Open System Environment (1996)

ISO: ISO/TC211, ISO 19119 Geographic information-service (2005)

ISO: ISO/TC211, ISO 19123 Geographic information – Schema for coverage geometry and functions (2005)

ISO/TC211 (2007) Geographic information/geomatics homepage. http://www.isotc211.org/. Accessed 2 Sept 2007

ISO/TC211: ISO19107:2003 Geographic Information—Spatial Schema. ISO, Geneva, Switzerland (2003)

ISO/TC211: ISO19108:2002 Geographic Information—Temporal Schema. ISO, Geneva, Switzerland (2002)

ISO/TC211: ISO19109:2005 Geographic Information—Rules for Application Schema. ISO, Geneva, Switzerland (2005)

ISO/TC211: ISO19110:2005 Geographic Information—Methodology for Feature Cataloguing. ISO, Geneva, Switzerland (2005)

ISO/TC211: ISO19111:2002 Geographic Information—Spatial Referencing by Coordinates. ISO, Geneva, Switzerland (2002)

ISO/TC211: ISO19112:2003 Geographic Information—Spatial Referencing by Geographic Identifier. ISO, Geneva, Switzerland (2003)

ISO/TC211: ISO19115:2003 Geographic Information—Metadata. ISO, Geneva, Switzerland (2003)

ISO/TC211: ISO19135 Geographic Information—Procedures for Item Registration. ISO, Geneva, Switzerland (2005)

ISO/TC211: ISO/TS19103:2005 Geographic Information—Conceptual Schema Language. ISO, Geneva, Switzerland (2005)

ISO/TR 19121: 2000. Geographic Information-Imagery and Gridded data. http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=29775

ISO/TS 19101-2. Geographic Information-Reference Model – Part 2: Imagery. http://www.isotc211.org/pow.htm

ISPRS Commission V Close-Range Sensing: Analysis and Applications. http://www.commission5.isprs.org/. Accessed 20 Aug 2007

ISPRS Working Group III/3: Processing of point clouds from laser scanners and other sensors. http://www.isprs.org/commission3/wg3/

ISPRS Working Group V/3: Terrestrial laser scanning. http://www.isprs.org/commission5/wg3/

Isserman, A.M.: The Location Quotient Approach for Estimating Regional Economic Impacts. J. Am. Inst. Plann. **43**, 33–41 (1977)

Istok, J.D., Cooper, R.M.: Geostatistics Applied to Groundwater Pollution. Journal of Environmental Engineering **114**, 915–928 (1988)

Iwasaki, K., Yamazawa, K., Yokoya, N.: An Indexing System for Photos Based on Shooting Position and Orientation with Geographic Database. In: Proceedings of IEEE international conference on multimedia and expo (ICME), 390–393 (2005)

Iwerks, G.S., Samet, H., Smith, K.P.: Maintenance of k-nn and spatial join queries on continuously moving points. ACM Transactions on Database Systems. **31**(2) (2006)

Jackson, M.J., Woodsford, P.A.: GIS data capture hardware and software. In: Geographical Information Systems, pp. 239–249. Longman Scientific and Technical, Harlow, England and Wiley, New York (1991)

Jacobsen, H.-A. Middleware for location-based services. In: Schiller, J., Voisard, A. (eds.). Location-Based Services. Morgan Kaufmann. Elsevier, pp. 83–114 (2004)

Jacobs, J.: The Death and Life of Great American Cities. Vintage Books, New York (1961)

Jacquez, G. M., Goovaerts, P., Rogerson, P.: Design and implementation of a space-time intelligence system for disease surveillance. J. Geogr. Syst. **7**, 7–23 (2005)

Jacquez, G.M.: Spatial Analysis in Epidemiology: Nascent Science or a Failure of GIS? J Geo Sys **2**, 91–97 (2000)

Jaffar, J., Lassez, J.L.: Constraint Logic Programming. In: Proceedings 14th ACM Symposium on Principles of Programming Languages, pp. 111–119 (1987)

Jagadish, H.V.: Linear clustering of objects with multiple attributes. In: Proc. of the intl. conf. on Management of data, SIGMOD, Atlantic City, NJ, June 1990, pp. 332–342

Jagadish, H.V.: On indexing line segments, VLDB '90: Proceedings of the 16th International Conference on Very Large Data Bases San Francisco, CA, USA, pp. 614–625, Morgan Kaufmann Publishers Inc. (1990)

Jagadish, H.V., Ooi, B.C., Tan, K.-L., Cui, Yu, Rui, Zhang: iDistance: An adaptive B+-tree based indexing method for nearest neighbor search. ACM Trans. Database Syst. 30(2):364–397 (2005)

Jagadish, H.V.: A retrieval technique for similar shapes. In: Proc. ACM International Conference on Management of Data (SIGMOD'91), pp. 208–217. ACM Press, New York, NY, USA (1991)

Jagoe, A.: Mobile Location Services, The Definitive Guide. Prentice Hall Publisher, Upper Saddle River, NJ (2003)

Jain, A.K.: Fundamentals of Digital Image Processing. Prentice Hall, New York, NY (1998)

Jain, L. C., Ghosh, A.:Evolutionary Computation in Data Mining. Springer, New York, NY (2005)

Jain, M.K., Kothyari, U.C., Ranga Raju, K.G.: A GIS Based Distributed Rainfall–runoff Model. J. Hydrol. **299**(1–2), 107–135 (2004)

Janeja V.P., Atluri, V.: FS^3: A random walk based free-form spatial scan statistic for anomalous window detection. In: ICDM, pp. 661–664 (2005)

Janeja V.P., Atluri, V.: ls^3: A linear semantic scan statistic technique for detecting anomalous windows. In: ACM Symposium on Applied Computing (2005)

Janelle, D.G.: Impact of information technologies. in Hanson, S., Giuliano, G. (eds.) The Geography of Urban Transportation, 3ed.. Guilford, 86–112 (2004)

Jankowski, P., Andrienko, N., Andrienko, G.: Map-centred exploratory approach to multiple criteria spatial decision making. Int. J. Geogr. Inf. Sci. **15**, 101–127 (2001)

Jankowski, P., Ewart, G.: Spatial decision support system for health practitioners: Selecting a location for rural health practice. Geographical Systems **3**(2), 297–299 (1996)

Jankowski, P., Nyerges, T.: Geographic information systems for group decision making: Towards a participatory geographic information science. Taylor and Francis, New York (2001)

Jankowski, P., Nyerges, T.L., Smith, A., Moore, T.J., Horvath, E.: Spatial group choice: A SDSS tool for collaborative spatial decision-making. International Journal of Geographical Information Systems **11**, 566–602 (1997)

de Janvry, A., Gordillo, G., Platteau, J.-P., Sadoulet, E. (eds.): Access to Land, Rural Poverty and Public Action. UNU/WIDER Studies in Development Economics. Oxford University Press, UK and Europe (2001) de Janvry, A., Sadoulet, E. (2001) Access to Land and Land Policy Reforms. Policy Brief No. 3, April 2001. http://www.wider.unu.edu/publications/pb3.pdf. Accessed 14 Aug 2007

Japanese Statistics Bureau, Ministry of Internal Affairs and Communications: Population Census, Daytime Population http://www.stat.go.jp/english/data/kokusei/2000/jutsu1/00/01.htm (2000). Accessed 17 Sept 2007

Jayant, N.S., Noll, P.: Digital coding of waveforms. Prentice-Hall, Englewood Cliffs, NJ (1984)

Jensen C.S.: Database Aspects of Location-Based Services, In: Schiller, J., Voisard, A.: Location-Based Services, Morgan Kaufmann Publishers, pp. 115–148 (2004)

Jensen, C.S., Lin, D., Ooi, B.C.: Query and Update Efficient B^+-Tree Based Indexing of Moving Objects. In: VLDB, Toronto, Canada, 29 Aug – 3 Sept 2004, pp. 768–779

Jensen, C.S., Tiešytė, D., Tradišauskas, N.: Robust B^+-Tree-Based Indexing of Moving Objects. In: Proceedings of the MDM Conference, p. 12, Nara, Japan, 9–13 May 2006

Jensen, C.S., Tiešytye, D., Tradišauskas, N.: Robust B+-tree-based indexing of moving objects, MDM '06: Proceedings of the 7th International Conference on Mobile Data Management (MDM'06) Washington, DC, USA, IEEE Computer Society, p. 12 (2006)

Jensen, J.R.: Remote Sensing of the Environment: An Earth Resource Perspective. Prentice-Hall, Upper Saddle River (2000)

Jensfelt, P.: Approaches to mobile robot localization in indoor environments. PhD thesis, Royal Institute of Technology, Stockholm (2001)

Jiang, B.: I/O-Efficiency of Shortest Path Algorithms: An Analysis. In: ICDE, pp. 12–19, Tempe, AZ, USA, 1992. IEEE Computer Society (1992)

Jiang, B., Yao, X.: Location-based services and GIS in perspective. Comput., Environ. Urban Syst. **30**, 712–725 (2006)

Jiang, H., Esatman, J.R.: Applications of fuzzy measures in multi-criteria evaluation in GIS. Int J Geogr Inf Sci **14**(2), 173–184 (2000)

Jing, N., Huang, Y.W., Rundensteiner, E.A.: Hierarchical Encoded Path Views for Path Query Processing: An Optimal Model and Its Performance Evaluation. TKDE **10**(3):409–432 (1998)

Jin, G., Nittel, S.: NED: Efficient Event Detection in Sensor Network. In: Proceedings of MDM-Workshop "Mobile Location-Aware Sensor Networks", Nara, Japan 2006

Jin, S., Bestavros, A.: Popularity-Aware Greedy Dual-Size Web Proxy Caching Algorithms. In: Proceedings of the International Conference on Distributed Computing Systems (ICDCS), Taipei, Taiwan, 10–13 Apr 2000, pp. 254–261

João, E.M.: Causes and Consequences of Map Generalization. Taylor and Francis Ltd., London (1998)

Johnson, M.: The Body in the Mind. Chicago Press, Chicago (1987)

Johnson, M.P.: Spatial decision support for assisted housing mobility counseling. Decis. Supp. Sys. **41**(1), 296–312 (2005)

Johnson, S.: The Ghost Map: The Story of London's Most Terrifying Epidemic—and How It Changed Science, Cities, and the Modern World. Riverhead, New York (2006)

Joint Center for Intelligent Spatial Computing at George Mason University (2005) http://landscan.scs.gmu.edu:8080/. Accessed 20 Aug 2007

Jones, C.B., Abdelmoty, A.I., Finch, D., Fu, G., Vaid, S.: The SPIRIT spatial search engine: architecture, ontologies and spatial indexing. In: Freksa, C., Egenhofer, M.J., Miller, H.J. (eds.) Proceedings of the Third International Conference on Geographic Information Science, Adelphi, MD, USA, 2004. Lecture Notes in Computer Science vol. 3234, pp. 125–139. Springer, USA (2004)

Jones, C.B.: Geographical Information Systems and Computer Cartography. Addison Wesley Longman, Harlow, Essex, UK (1997)

Jones, C.B.: Spatial information retrieval and geographical ontologies: an overview of the spirit project. In: Proceedings of ACM SIGIR conference on Reasearch and developement in information retrieval, Tampere, Finnland, 11 Aug 2002

Jones, C.B., Ware, J.M.: Map generalization in the web age. Int. J. Geogr. Inf. Sci. **19**, 859–870 (2005)

Jones, K.B., Riitters, K.H., Wickham, J.D., Tankersley Jr., R.D., O'Neill, R.V., Chaloud, D.J., Smith, E.R., Neale, A.C.: An Ecological Assessment of the United States Mid-Atlantic Region: A Landscape Atlas. U.S. EPA, Office of Research and Development. Washington, DC. EPA/600/R-97/130 (1997)

Jones, P.M., et al.: Coraven: modeling and design of a multimedia intelligent infrastructure for collaborative intelligence analysis. IEEE Trans. Syst., Man, Cybernet. **1**, 914–919 (1998)

de Jong, S.M., van der Meer, F.D.: Remote Sensing Image Analysis Including the Spatial Domain (Remote Sensing and Digital Image Processing). Springer, (2006)

Jordan, M.I.(ed.): Learning in Graphical Models. MIT Press, Cambrige, MA (1998)

Journel, A.G., Huijbregts, C.J.: Mining Geostatistics. Academic Press, New York (1992)

Juels, A., Pappu, R.: Financial Cryptography, In: Squealing Euros: Privacy Protection in RFID-Enabled Banknotes, Lecture Notes in Computer Science, vol. 2742, pp. 103–121. Springer (2003)

Juels, A., Rivest, R.L., Szydlo, M.: The blocker tag: selective blocking of RFID tags for consumer privacy, In: CCS '03: Proceedings of the 10th ACM conference on Computer and communications security, pp. 103–111. ACM Press, New York (2003)

Juels, A., Weis, S.A.: Defining Strong Privacy for RFID. http://eprint.iacr.org/2006/137.pdf, Cryptology ePrint Archive: Retreived 10/10/2006 (2006)

Jukic, N.: Modeling strategies and alternatives for data warehousing projects. Commun. ACM. **49**(4), 83–88 (2006)

Jurvetson, S.: What exactly is viral marketing? Red Herring **78**, 110–112 (2000)

Kaemarungsi, K.: Design of indoor positioning systems based on location fingerprinting technique. PhD thesis, University of Pittsburgh (2005)

Kaemarungsi, K., Krishnamurthy, P.: Properties of indoor received signal strength for WLAN location fingerprinting. In: Proceedings of the The First Annual Int. Conf. on Mobile and Ubiquitous Systems: Networking and Services (MOBIQUITOUS), Boston, USA, pp. 14–23. IEEE, Piscataway, NJ (2004)

Kahveci, T., Singh, A., Gurel, A.: Similarity Searching for Multi-attribute Sequences. In: IEEE Press, p. 175 (2002)

Kahveci, T., Singh, A.: Variable length queries for time series data. In: Proceedings of the 17th International Conference on Data Engineering, Heidelberg, Germany, 2–6 April 2001, pp. 273–282

Kain, R.J.P., Baigent, E.: The Cadastral Map in the Service of the State: A History of Property Mapping. The University of Chicago Press, Chicago (1993)

Kaitala, S., Shavykin, A., Volkov, V.A.: Environmental GIS database for the white sea. In: Proceedings of the Open source GIS – GRASS users conference, 11–13 September 2002, Trento, Italy (2002)

Kajiyama, A., Ikawa, N., Masumoto, S., Shiono, K., Raghavan, V.: Three-dimensional geological modeling by foss grass

GIS – using some field survey data –. In: Proceedings of the FOSS/GRASS Users Conference – Bangkok, 12–14 September 2004, Bangkog, Thailand (2004)

Kalashnikov, D.V., Prabhakar, S., Hambrusch S.E., Aref, W.A.: Efficient evaluation of continuous range queries on moving objects. In: DEXA '02: Proceedings of the 13th International Conference on Database and Expert Systems Applications, pp. 731–740. Springer, Heidelberg (2002)

Kalfoglou, Y., Hu, B., Reynolds, D., Shadbolt, N.: Semantic Integration Technologies Survey. CROSI Project, University of Southampton, Technical Report, E-Print No. 10842 May (2005)

Kalin, L., Hantush, M.M.: Evaluation of sediment transport models and comparative application of two watershed models (U.S. Environmental Protection Agency, National Risk Management Research Laboratory Paper EPA/600/R-03/139). U.S. Environmental Protection Agency, Cincinnati, OH (2003)

Kalnay, E.: Atmospheric Modeling, Data Assimilation and Predictability. Cambridge University Press (2004)

Kalnis, P., Ghinta, G., Mouratidis, K., Papadias, D.: Preserving anonymity in location based services. Technical report TRB6/06, University of Singapore (2006)

Kalnis, P., Mamoulis, N., Bakiras, S.: On discovering moving clusters in spatio-temporal data. In: Medeiros, C.B., Egenhofer, M.J., Bertino, E. (eds.) Proceedings of the 9th International Symposium on Advances in Spatial and Temporal Databases. Lecture Notes in Computer Science, vol. 3633, pp. 364–381. Springer, Berlin, Heidelberg (2005)

Kalnis, P., Mamoulis, N., Bakiras, S.: On discovering moving clusters in spatio-temporal data.In: Proceedings of the 9th International Symposium on Spatial and Temporal Databases (SSTD), pp. 364–381, Angra dos Reis, Brazil, August 2005

Kalnis, P., Ng, W.S., Ooi, B.C., Papadias D., Tan K.L.: Proxy-server architectures for OLAP. In: Proceedings of ACM-SIGMOD International Conference on Management of Data, Madison, Wisconsin, 3–6 June 2002, pp. 25–36

Kalnis, P., Papadias, D.: Proxy-Server Architectures for OLAP. In: Proceedings of ACM-SIGMOD International Conference on Management of Data, Santa Barbara, California, USA, 21–24 May 2001, pp. 367–378

Kalpakis, K., Gada, D., Puttagunta, V.: Distance measures for effective clustering of ARIMA time-series. In: Proceedings of the IEEE International Conference on Data Mining, San Jose, 29 November–2 December 2001, pp. 273–280

Kamel, I., Faloutsos, C.: Hilbert r-tree: An improved r-tree using fractals. In: Proc. of Very Large Data Bases (VLDB), Santiago de Chile, Chile, 12–15 Sept 1994, pp. 500–509

Kamel, I., Faloutsos, C.: On packing r-trees. In: Proc. of Conference on Information and Knowledge Management (CIKM), Washington DC, USA, 1–5 Nov 1993, pp. 490–499

Kamel, I., Faloutsos, C.: Parallel R-Trees. In: Proc. of ACM SIGMOD Conf., pp. 195–204. San Diego, CA, June 1992

Kamoun, F., Kleinrock, L.: Hierarchical Routing for Large Networks: Performance Evaluation and Optimization. Comput Netw **1**:155–174 (1977)

Kampshoff, S.: First International Workshop on Next Generation 3D City Models. Bonn (2005) http://www.gia.rwth-aachen.de/Service/document_download.php?fileTag=kambonn05

Kanaan, M., Akgül, F.O., Alavi, B., Pahlavan, K.: Performance benchmarking of TOA-based UWB indoor geolocation systems using MSE profiling. In: Proceedings of the IEEE Vehicular Technology Conference, Montreal, Canada, 25–28 Sept 2006

Kanaan, M., Pahlavan, K.: CN-TOAG: a new algorithm for indoor geolocation. In: Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Barcelona, 5–8 Sept 2004

Kanellakis, P.C., Kuper, G.M., Revesz, P.: Constraint query languages. J. Comput. Syst. Sci. **51**(1), 26–52 (1995)

Kanellakis, P.C., Kuper, G.M., Revesz, P.: Constraint Query Languages. In: Proc. ACM Symposium on Principles of Database Systems, pp. 299–313, Nashville, Tennessee, 1990

Kangas, J., Kohonen, T.: Development and applications of self-organizing map and related algorithms. Mathematics and Computers in Simulations **41**, 3–12 (1996)

Kang, I., Kim, T., Li, K.: A spatial data mining method by delaunay triangulation. In: 5th ACM int. workshop on Advances in GIS, pp. 35–39 (1997)

Kang, J.M., Mokbel, M.F., Shekhar, S., Xia, T., Zhang, D.: Continuous evaluation of monochromatic and bichromatic reverse nearest neighbors. In: ICDE (2007)

Kang-tsung Chang: Introduction to Geographic Information Systems, 4th edn., 101 (2008)

Kanjamala, P., Revesz, P., Wang, Y.: MLPQ/GIS: A GIS using linear constraint databases. Proc. 9th COMAD International Conference on Management of Data, pp. 389–393. McGraw-Hill, New Delhi (1998)

Kanoulas, E., Du, Y., Xia, T., Zhang, D.: Finding Fastest Paths on A Road Network with Speed Patterns. ICDE, Atlanta, GA, USA, 2006. IEEE Computer Society (2006)

Kantarcioglu, M., Clifton, C.: Privacy-preserving distributed mining of association rules on horizontally partitioned data. IEEE Trans. Knowl. Data Eng. **16**, 1026–1037 (2004)

Kaplan, E.D.: Understanding GPS. Artech, Boston (1996)

Kapler, T., Wright, W.: Geotime information visualization. In: InfoVis'04, pp. 25–32 (2004)

Karara, H.M. (ed.): Non-Topographic Photogrammetry. 2nd edn. American Society for Photogrammetry and Remote Sensing. Falls Church, Virginia (1989)

Karhunen, H.: Über lineare Methoden in der Wahrscheinlichkeitsrechnung. Ann. Acad. Science Fenn (1947)

Karlsson, J., Ronnqvist, M., Frisk, M.: RoadOpt: A decision support system for road upgrading in forestry. Scand. J. Forest Res. **21**(7), 5–15 (2006)

Karp, B., Kung, H.T.: Greedy perimeter stateless routing for wireless networks. In: Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000), pp. 243–254: Boston, MA (2004)

Karras, P., Mamoulis, N.: One-pass wavelet synopses for maximum-error metrics. In: VLDB '05: Proceedings of the 31st international conference on Very large data bases, pp. 421–432. VLDB Endowment, Trondheim, Norway (2005)

Karssenberg, D., De Jong, K.: Dynamic environmental modelling in GIS. 2: Modelling error propagation. Int. J. Geogr. Inf. Sci. **19**, 623–637 (2005)

Karuppannan, J.: Mapping and Corrections: Management of Offenders with Geographic Information Systems. Corrections Compendium. http://www.iaca.net/Articles/drjaishankarmaparticle.pdf (2005)

Karypis, G., Han, E.H., Kumar, V.: CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. IEEE Comput. **32**(8), 68–75, (1999)

Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. In: Intern. J. Comput. Vis.**1**(4), 321–331 (1987)

Kaufman, L., Rousseeuw, P.J.: Finding groups in data: an introduction to cluster analysis. Wiley series in probability and mathematical statistics. Applied probability and statistics.: Wiley, New York, xiv, 342 p (1990)

Kaufmann, J., Steudler, D. (1998) Cadastre 2014. http://www.fig.net/cadastre2014/index.htm. Accessed 14 Aug 2007

Kaufman, S.: A formal theory of spatial reasoning. In: Proc Int. Conf. on Knowledge Representation and Reasoning, pp. 347–356, Morgan Kaufmann, San Mateo (1991)

Kay, S.M.: Fundamentals of Statistical Signal Processing: Estimation Theory. Prentice Hall, Upper Saddle River, NY (1993)

Kazman, R., Kominek, J.: Information organization in multimedia resources. Proceedings of the 11th annual international conference on Systems documentation, pp149–162 (1993)

Keenan, P.B.: Spatial Decision Support Systems. In: Mora, M., Forgionne, G., Gupta, J.N.D. (eds.) Decision Making Support Systems: Achievements and Challenges for the New Decade. pp. 28–39. Idea Group, Harrisburg, PA (2003)

Keeney, R.L., Raffia, H.: Decisions with multiple objectives: Preferences and value trade-offs. John Wiley & Sons, New York (1976)

Keeney, R.L.: Valued-focused thinking: A path to creative decision. Harvard University Press, Cambridge, USA (1992)

Keil, J.M.: Decomposing a Polygon into Simpler Components. SIAM Journal of Computing **14**(4), 799–817 (1985)

Keim, D.: Efficient geometry-based similarity search of 3D spatial databases. In: Proc. ACM International Conference on Management of Data (SIGMOD'99), pp. 419–430. ACM Press, New York, NY, USA (1999)

Kelejian, H., Prucha, I.: Prediction Efficiencies in Spatial Models with Spatial Lags, University of Maryland (2004)

Keller, P.R., Keller, M.M.: Visual Cues, Practical Data Visualization. IEEE, Piscataway (1992)

Kelsall, J., Diggle, P.: Spatial variation in risk of disease: a nonparametric binary regression approach. Appl. Stat. **47**, 559–573 (1998)

Kelsall, J., Wakefield, J.: Modeling spatial variation in disease risk: a geostatistical approach. J. Am. Stat. Assoc. **97**, 692–701 (2002)

Kempe, D., Kleinberg, J.M., Demers, A.J.: Spatial gossip and resource location protocols. J. ACM **51**(6), 943–967 (2004)

Kempf, K.: Historical Monograph: Electronic Computers Within the Ordnance Corps (1961) Available via: http://ed-thelen.org/comp-hist/U-S-Ord-61.html. Accessed 29 Aug 2007

Kendall, M.G.: A Course in the Geometry of *n* Dimensions. Hafner, New York (1961)

Kenyon, S.H.: Regional Effort Forges Emergency Coordination System, Signal Magazine (2004). http://www.afcea.org/signal/articles/anmviewer.asp?a=13z=2. Accessed Aug 2007

Keogh, E., Chakrabarti, K., Mehrotra, S., Pazzani, M.: Locally adaptive dimensionality reduction for indexing large time series databases. In Proc. of ACM SIGMOD, pp. 151–162 (2001)

Keogh, E.: Exact indexing of dynamic time warping. In Proc. of VLDB (2002)

Keogh, E.J., Chu, S., Hart, D., Pazzani, M.J.: An online algorithm for segmenting time series. In: ICDM, pp. 289–296 (2001)

Keogh, E., Kasetty, S.: On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. In: The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, 23–26 July 2002, pp. 102–111

Keogh, E., Lonardi, S., Chiu, W.: Finding Surprising Patterns in a Time Series Database In Linear Time and Space. In: The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, 23–26 July 2002, pp. 550–556

Keogh, E., Pazzani, M.: An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In: Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, New York, 27–31 August 1998, pp. 239–241

Keogh, E., Pazzani, M.: An Indexing Scheme for Fast Similarity Search in Large Time Series Databases. In: IEEE Press, 56–67 (1999)

Kephart, J.O., White, S.R.: Directed-graph epidemiological models of computer viruses. Proc. 1991 IEEE Computer Society Symposium on Research in Security and Privacy, pp. 343–359 (1991)

Keramitsoglou, I., Cartalis, C., Kassomenos, P.: Decision support system for managing oil spill events. Env. Manag. **32**(2), 290–298 (2003)

Kersting, O., Dollner, J.: Interactive 3D visualization of vector data in GIS. In: Proceedings of the 10th ACM International Symposium on Advances in Geographical Information Systems, pp. 107–112. McLean, Virginia (2002)

Khan, L., Awad, M., Thuraisingham, B.: A new intrusion detection system using support vector machines and hierarchical clustering. VLDB J. (2007, in press)

Khatri, V., Ram, S., Snodgrass, R.T.: On Augmenting Database Design-Support Environments to Capture the Geo-Spatio-temporal Data Semantics. In: Information Systems, Vol.31, No.2, pp. 98–133. Elsevier Science Ltd., Oxford, UK (2006)

Khedker, U., Dhamdhere, D.M.: A generalized theory of bit vector data flow analysis. ACM Trans. Programming Languages Syst. **16**, 1472–1511 (1994)

Kibria, B.M.G., Sun, L., Zidek, J.V., Le, N.D.: Bayesian spatial prediction of random space-time fields with application to mapping $PM_{2.5}$ exposure. J. Am. Stat. Assoc. **97**, 112–124 (2002)

Kiehle, C., Greve, K. Standardized Geoprocessing – Taking Spatial Data Infrastructures One Step Further. In: Proceedings of AGILE, Minneapolis, Minnesota, USA, 23–28 July 2006

Kiely, P.M., Smith, G., Carney, L.G.: The mean shape of the human cornea. J. Modern Opt. **29**(8), 1027–1040 (1982)

Kilpelainen, T., Sajakoski, T.: Incremental generalization for multiple representations of geographical objects. In: Muller, J.C., Lagrange, J.P., Weibel, R. (eds.) GIS and Generalization: Methodology and Practice, pp. 209–218. Taylor & Francis, London (1995)

Kimball, R.: The Data Warehouse Toolkit, 2nd edn. Wiley, New York (2002)

Kimball, R.: The Data Warehouse Toolkit. John Wiley, USA (1996)

Kim, S., Pals, T., Iltis, R.A., Lee, H.: CDMA multipath channel estimation using generalized successive interference cancellation algorithm for radiolocation. In: 36th Conf. Information Sciences & Systems, March (2002)

Kim, S., Shekhar, S.: Contraflow network reconfiguration for evacuation planning: A summary of results. In: Proceedings of the 13th ACM Symposium on Advances in Geographic Information Systems. pp. 250–259 (2005)

King, L.J.: Statistical Analysis in Geography. Prentice Hall, Englewood Cliffs, NJ (1969)

Kingston, R., Turton, I., Carver, S., Evans, A.: Web-based public participation geographical information systems: An aid to local environmental decision-making. Comput. Environ. Urban Syst. **24**, 109–125 (2000)

Kirkpatrick, S., Gelatt, C.D., Jr., Vecchi, M.P.: Optimization by simulated annealing. Science **220**, 671–680 (1983)

KLD (1094): Formulations of the DYNEV and I-DYNEV Traffic Simulation Models Used in ESF. Federal Emergency Management Agency

KLD: Development of IDYNEV and PCDYNEV. http://www.kldasociates.com/evac.htm (2004)

Klein R., Wood, D.: Voronoi diagrams based on general metrics in the plane. In: Proceedings of the 5th Annual Symposium on Theoretical Aspects of Computer Science, pp. 281–291 (1998)

Kleppe, A. (2004): Interview with Anneke Kleppe. Code Generation Network. http://www.codegeneration.net/tiki-read_article.php?articleId=21. date cited: August 2006

Kleppe, A., Warmer, J., Bast, W.: MDA Explained: The Model Driven Architecture-Practice and Promise. Addison-Wesley Professional, London (2003)

Klosterman, R.E.: (1999) The What If? Collaborative Planning Support System. Environment and Planning B: Planning and Design **26**:398–408 (1999) See also http://www.what-if-pss.com

Klosterman, R.E.: Computer applications in planning. Environ. Plann. B: Plann. Des. **25**, 32–36 (1998)

Klosterman, R.E., Pettit C.J.: Guest editorial: an update on planning support systems. Environment and Planning B: Planning and Design **32**:477–484 (2005) London

Klosterman, R.E., Siebert, L., Hoque, M.A., Kim, J., Parveen, A.: Using an operational planning support system to evaluate farmland preservation policies. In: Geertman, S., Stillwell, J. (eds.) Planning Support Systems in Practice, pp. 391–407. Springer, New York, NY (2003) see also What if? at http://www.what-if-pss.com

Klosterman, R.E.: The What if? planning support system. In: Brail, R.K., Klosterman, R.E. (eds.) Planning Support Systems: Integrating geographic information systems, models, and visualization tools, pp. 263–284. ESRI Press (2001) See also http://www.what-if-pss.com

K-M Choi, Christakos, G., Wilson, M.L.: El Niño effects on influenza mortality risks in the state of California. J. Public Health **120**, 505–516 (2006)

Kneib, T., Fahrmeir, L.: A Mixed Model Approach for Geoadditive hazard Regression. Scandinavian J. Stat. **34**, 207–228 (2007)

Knorr-Held, L.: Bayesian modelling of inseparable space-time variation in disease risk. Stat. Med. **19**, 2555–2567 (2000)

Koelle, A.R., Depp, S.W., Freyman, R.W.: Short-range radiotelemetry for electronic identification, using modulated RF backscatter, Proc. IEEE **63**(8), 1260–1261 (1975)

Koestler, A.: The Ghost in the Machine. Pan, London (1967)

Kohler, E., Langtau, K., Skutella, M.: Time-Expanded Graphs for Flow-Dependent Transit Times. In: Proceedings of 10th Annual European Symposium on Algorithms (2002)

Kohonen T: (2001) SOM-Tool Box Home. http://www.cis.hut.fi/projects/somtoolbox/. cited 02.04.01

Kohonen, T.: The self-organizing map. IEEE **78**, 1464–1480 (1990)

Kohonen, T.: Self-organizing Maps. Springer-Verlag, Inc., New York (1995)

Kokla, M., Kavouras, M.: Fusion of top-level and geographical domain ontologies based on context formation and complementarity. Int. J. Geogr. Inf. Sci. **15**(7), 679–687 (2001)

Kolahdouzan, M., Shahabi, C.: Continuous K-Nearest Neighbor Queries in Spatial Network Databases. STDBM (2004)

Kolahdouzan, M., Shahabi, C.: Voronoi-Based K Nearest Neighbor Search for Spatial Network Databases, In: Proc. International Conference on Very Large Data Bases, pp. 840–851 (2004)

Kolesnikov, A.: Efficient algorithms for vectorization and polygonal approximation. PhD thesis, University of Joensuu, Finland, (2003)

Kollios, G., Gunopulos, D., Tsotras, V.J.: On indexing mobile objects. PODS '99: Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. pp. 261–272 ACM Press, New York, NY, USA (1999)

Kollios, G., Papadopoulos, D., Gunopulos, D., Tsotras, V.J.: Indexing Mobile Objects Using Dual Transformations. The VLDB Journal **14**(2):238–256 (2005)

Kollios, G., Tsotras, V.J., Gunopulos, D., Delis, A., Hadjieleftheriou, M.: Indexing animated objects using spatiotemporal access methods. TKDE **13**(5), 758–777 (2001)

Kolmogorov, A.N.: Interpolation and extrapolation of stationary random sequences. Isvestiia Akademii Nauk SSSR, Seriia Matematicheskiia **5**, 3–14 (1941) (Translation, 1946, Memo RM-3090-PR, Rand Corp., Santa Monica, CA)

Kolodziej, K., Hjelm, J.: Local Positioning Systems, LBS Applications and Services. Taylor and Francis Group, CRC Press, Boca Raton, FL (2006)

Kolovos, A., Christakos, G., Serre, M.L., Miller, C.T.: Computational BME solution of a stochastic advection-reaction equation in the light of site-specific information. Water Res. Res. **38**, 1318–1334 (2002)

Kolovson, C., Stonebraker, M.: Segment Indexes: Dynamic indexing techniques for multi-dimensional interval data. In: Proc. of ACM Management of Data (SIGMOD),Colorado, USA, 29–31 May 1991, pp. 138–147

Komarov, A. S., Palenova, M.M., Smirnova, O.V.: The concept of discrete description of plant ontogenesis and cellular automata models of plant populations. Ecol. Modelling **170**, 427–439 (2003)

Komzak, J., Slavik, P.: Architecture of system for configurable GIS data compression. http://kmi.open.ac.uk/people/jiri/publ/wscg2002.pdf

Koncz, N., Adams, T.M.: A Data Model for Multi-Dimensional Transportation Location Referencing Systems, J. Urban Reg. Inform. Syst. Assoc. **14**(2), 27–41 (2002)

Konks D, and Challa S (2005) An Introduction to Bayesian and Dempster-Shafer Data Fusion Available via DSTO-TR-1436, Edinburgh, Australia, November 2005. http://www.dsto.defence.gov.au/publications/2563/DSTO-TR-1436.pdf

Kooistra, L., Huijbregts, M.A.J., Ragas, A.M.J., Wehrens, R., Leuven, R.S.E.W.: Spatial Variability and Uncertainty in Ecological Risk Assessment: A Case Study on the Potential Risk of Cadmium for the Little Owl in a Dutch River Flood Plain. Environ. Sci. Technol. **39**, 2177–2187 (2005)

Koo, S.G.M., Rosenberg, C., Chan, H-H, Lee, Y.C.: Location-based e-campus web services: from design to deployment. In: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, Fort Worth, Texas, USA 23–26 March 2003

Koperski, K., Adhikary, J., Han, J.: Spatial data mining: Progress and challenges. In: Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'96), p. 1–10, Montreal, Canada June 1996

Koperski, K., Han, J.: Discovery of spatial association rules in geographic information databases. In SSD 95: Proceedings of the 4th International Symposium on Advances in Spatial Databases, pp. 47–66. Springer-Verlag (1995)

Koppell, I.: What are Location Services? – From a GIS Perspective, LBS360.NET (http://www.lbs360.net/LBSArticles/ESRI.What%20are%20LS%20Whitepaper.pdf)

Kornacker, M., Banks, D.: High-Concurrency Locking in R-Trees. In: Proceedings of International Conference on Very Large Data Bases, Zurich, Switzerland, 11–15 Sept 1995

Kornacker, M., Mohan, C., Hellerstein, J.: Concurrency and Recovery in Generalized Search Trees. In: Proceedings of ACM SIGMOD International Conference on Management of Data, Tucson, Arizona, 13–15 May 1997

Korn, F., Muthukrishnan, S.: Influence Sets Based on Reverse Nearest Neighbor Queries. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of data. pp. 201–212, ACM Press (2000)

Kossmann, D., Ramsak, F., Rost, S.: Shooting Stars in the Sky: An Online algorithm for Skyline Queries. In: Proceedings of VLDB'02, pp. 275–286 Hong Kong, China (2002)

Kotch, T.: Cartographies of Disease: Maps, Mapping and Disease. ESRI Press, Redlands, CA (2005)

Kothuri, R., Godfrind, A., Beinat, E.: Pro Oracle Spatial. Apress, Berkley, CA, USA (2004)

Kothuri, R., Ravada, S.: Efficient processing of large spatial queries. In: Proceedings of the Symposium of Spatial and Temporal Databases, Redondo Beach, CA, 12–15 July 2001

Kothuri, R., Ravada, S., Sharma, J., Banerjee, J.: Indexing medium dimensionality data in Oracle. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. Philadelphia, PA, 1–3 June 1999

Kotidis, Y., Roussopoulos, N.: DynaMat: A dynamic view management system for data warehouses. In: Proceedings of ACM-SIGMOD International Conference on Management of Data, Philadelphia, Pennsylvania, USA, 1–3 June 1999, pp. 371–382

Koua, E., MacEachren, A.M., Kraak, M.J.: Evaluating the usability of an exploratory geovisualization environment. Int. J. of Geogr. Inf. Sci. **20**, 425–448 (2006)

Koubarakis, M. et al. (eds.): Spatio-temporal Databases: The Chorochronos Approach. Lecture Notes in Computer Science, vol. 2520. Springer, Berlin, Heidelberg (2003)

Koubarakis, M., Sellis, T., Frank, A.U., Grumbach, S., Güting, R.H., Jensen, C.S., Lorentzos, N.A., Manolopoulos, Y., Nardelli, E., Pernici, B., Schek, H.J., Scholl, M., Theodoulidis, B., N. Tryfona, N. (eds.): Spatio-temporal Databases: The CHOROCHRONOS Approach. Lecture Notes in Computer Science 2520, Springer, Berlin (2003) ISBN 3-540-40552-6.

Koubarakis, M., Sellis, T.K., Frank, A.U., Grumbach, S., Güting, R.H., Jensen, C.S., Lorentzos, N.A., Manolopoulos, Y., Nardelli, E., Pernici, B., Schek, H.-J., Scholl, M., Theodoulidis, B., Tryfona, N.: Spatio-Temporal Databases: The CHOROCHRONOS Approach. Springer, New York (2003)

Koudas, N., Faloutsos, C., Kamel, I.: Declustering Spatial Databases on a Multi-Computer Architecture. In: International Conference on Extending Database Technology (EDBT), pp. 592–614. Avignon, France, 25–29 March 1996

Kou, Y., Lu, C., D. Chen, D.: Spatial weighted outlier detection. In: Proceedings of the 6th SIAM Int. Conference on Data Mining 2006, Bethesda, MD, USA, 20–22 April (2006)

Kovar, K., Nachtnebel, H.P.: Application of geographic information systems in hydrology and water resources management. In: Proceedings of the HydroGIS'96 Conference, Vienna, 16–19 Apr 1996

Ko, Y.-B., Vaidya, N.H.: Location-aided routing (LAR) in mobile ad hoc networks. Wireless Networks. **6**(4) (2000)

Ko, Y., Vaidya, N.H.: Location-aided routing (LAR) in mobile ad hoc networks. In: Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking, 1998 (MOBICOM '98), Dallas, Texas, USA, 25–30 Oct 1998

Kraak, M.-J., Brown, A. (eds.) Web Cartography – Developments and prospects, Taylor & Francis, New York, ISBN 0-7484-0869-X (2001)

Kraak, M.J., Ormeling, F.J.: Cartography, Visualization of Geospatial Data, 2nd edn. Prentice Hall, London (2003)

Kraak, M.-J.: Settings and needs for web cartography, In: Kraak, Brown, A (eds.) Web Cartography. p. 3–4, Taylor & Francis, New York (2001)

Kraak, M.J., Vlag, D.v.d.:7 Understanding Spatiotemporal Patterns: Visual Ordering of Space and Time. Cartographica **42**(2), 153–162 (2007)

Krabill, W.B., Collins, J.G., Link, L.E., Swift, R.N., Butler, M.L.: Airborne laser topographic mapping results. Photogramm. Eng. Remote Sensing **50**, 685–694 (1984)

Kremer, J.: CCNS and AEROcontrol: Products for efficient photogrammetric data collection. Fritsch/Spiller (eds.) Photogrammetric Week 2001, Wichmann Verlag, Heidelberg, Germany, pp. 85–92. (2001)

Kriegel, H., Potke, M., Seidl, T.: Managing intervals efficiently in object-relational databases. In: VLDB, 2000

Krishnamachari, B., Estrin, D., Wicker, S.: The impact of data aggregation in wireless sensor networks. In: Proceedings of the 22nd International Conference on Distributed Computing Systems, Workshops, Vienna, Austria, 2–5 July 2002

Krivoruchko, K., Gotway, C.A.: Assessing the Uncertainty Resulting from Geoprocessing Operations. submitted for publication, ESRI Press, http://www.esri.com/library/whitepapers/pdfs/assessing-uncertainty.pdf (2003)

Krolak, P., Felts, W., Marble, G.: A man-machine approach toward solving the traveling salesman problem. Communications of the ACM **14**:(5), (1971)

Kros, J., Pebesma, E.J., Reinds, G.J., Finke, P.F.: Uncertainty assessment in modelling soil acidification at the European scale: a case study. J. Environ. Qual. **28**, 366–377 (1999)

Krug, K., Mountain, D., Phan, D.: Location-based services for mobile users in protected areas. Geoinformatics, **6**(2), 26–29 (2003)

Krumm, J., Cermak, G., Horvitz, E.: Rightspot: A novel sense of location for a smart personal object. In: Fifth International Conference on Ubiquitous Computing, October (2003)

Kuhn, F., Wattenhofer, R., Zhang, Y., Zollinger, A.: Geometric adhoc routing: of theory and practice. PODC '03: Proceedings of the 22nd Annual Symposium on Principles of Distributed Computing, Boston, MA, 13–16 July 2003

Kuhn, W.: Ontologies in support of activities in geographical space. Int. J. Geogr. Inf. Sci. **15**, 613–631 (2001)

Kuikka, S., Varis, O.: Uncertainties of climate change impacts in Finnish watersheds: a Bayesian network analysis of expert knowledge. Boreal Environ. Res. **2**, 109–128 (1997)

Kukich, K.: Techniques for automatically correcting words in text. ACM Computing Surveys **24**(4), 377–440 (1992)

Kulik, J., Heinzelman, W., Balakrishnan, H.: Negotiation-based protocols for disseminating information in wireless sensor networks, Wireless Networks, vol. 8, pp. 169–185 (2002)

Kulik, L.: A geometric theory of vague boundaries based on supervaluation. In: Montello, D.R. (ed.): Spatial Information Theory: Foundations of Geographic Information Science, International Conference, COSIT 2001, Morro Bay, CA, USA, September 19–23, 2001, Proceedings, volume 2205 of Lecture Notes in Computer Science, pp. 44–59. Springer, Heidelberg (2001)

Kulik, L., Tanin, E., Umer, M.: Efficient data collection and selective queries in sensor networks. In: Nittel, S., Labrinidis, A., Stefanidis, A. (eds.) Advances in Geosensor Networks. Springer, New York (2007, in press)

Kulldorff, M., Feuer, E., Miller, B., Freedman, L.: Breast cancer in northeast United States: a geographic analysis. Am. J. Epidemiol. **146**, 161–170 (1997)

Kulldorff, M., Nagarwalla, N.: Spatial disease clusters:detection and inference. Stat. Med. **14**, 799–810 (1995)

Kulldorff, M.: A spatial scan statistic. Commun. Stat Theor. Methods **26**, 1481–1496 (1997)

Kulldorff, M., Tango, T., Park, P.J.: Power comparisons for disease clustering tests. Comput. Stat. Data Anal. **42**, 665–684 (2003)

Kulyukin, V., Gharpure, C., Nicholson, J., Pavithran, S.: RFID in robot-assisted indoor navigation for the visually impaired. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004, pp. 1979–1984. (2004)

Kumar, A., Tsotras, V.J., Faloutsos, C.: Designing access methods for bitemporal databases. IEEE Transactions Knowl. Data Eng. (TKDE) **10**(1):1–20 (1998)

Kumar, V., Karande, K.: The effect of retail store environment on retailer performance. J. Bus. Res. **49**(2), 167–181 (2000)

Kung, R.M., Hanson, E.N., Ioannidis, Y.E., Sellis, T.K., Shapiro, L.D., Stonebraker, M.: Heuristic Search in Data Base Systems. In: Expert Database Systems Workshop (EDS), pp. 537–548 (1984)

Kunsch, H.R.: Intrinsic autoregressions and related models on the two-dimensional lattice. Biometrika. **74**, 517–524 (1987)

Kuper, G.M., Libkin, L., Paredaens, J. (eds.): Constraint Databases. Springer-Verlag, Berlin/Heidelberg (2000)

Kushki, A., Plataniotis, K., Venetsanopoulos, A.: Kernel-based positioning in wireless local area networks. IEEE Trans. on Mobile Computing **6**(6), 689–705 (2007)

Kushki, A., Plataniotis, K., Venetsanopoulos, A.N.: Location tracking in wireless local area networks with adaptive radio maps. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Toulouse, France, vol. 5, pp. 741–744. IEEE, Piscataway, NJ (2006)

Ku, W.-S., Haojun, W., et al.: ANNATO: Adaptive Nearest Neighbor Queries in Travel Time Networks. 7th International Conference on Mobile Data Management 2006. pp. 50. May 10–12 (2006)

Kwan, M.-P.: GIS Methods in time-geographic research: geocomputation and geovisualization of human activity patterns. Geogr. Ann., Ser. B Hum. Geogr. **86**, 267–280 (2004)

Kwan, M.P.: Interactive geovisualization of activity-travel patterns using three dimensional geographical information systems: a methodological exploration with a large data set. Transportation Research Part C **8**(1–6), 185–203 (2000)

Kwartler, M., Bernard, R.N.: CommunityViz: an integrated planning support system. In: Brail, R.K., Klosterman, R.E. (eds.) Planning Support Systems: integrating geographic information systems, models, and visualization tools, pp. 285–308. ESRI Press (2001) see also http://www.communityviz.com

Kwon, D., Lee, S., Lee, S.: Indexing the current positions of moving objects using the lazy update R-tree. In: Proc. MDM, pp. 113–120 (2002)

Kwon, E., Pitt, S.: Evaluation Of Emergency Evacuation Strategies For Downtown Event Traffic Using a Dynamic Network Model. Presented at Annual Meeting Of Transportation Research Board, Washington, D.C. (2005)

Kwon, E., Pitt, S.: Evaluation of Emergency Evacuation Strategies for Downtown Event Traffic Using a Dynamic Network Model. Transp. Res. Record: J. Transp. Res. Board **1922**, 149–155 (2005)

Kyriakidis, P.C., Dungan, J.L.: A geostatistical approach for mapping thematic classification accuracy and evaluating the impact of inaccurate spatial data on ecological model predictions. Environ. Ecol. Stat. **8**, 311–330 (2001)

Kyriakidis, P.C., Journel, A.G.: Geostatistical space-time models: a review. Math. Geol. **31**, 651–684 (1999)

Kyriakidis, P., Goodchild, M.F.: On the prediction error variance of three common spatial interpolation schemes. Int. J. Geogr. Inf. Sci. **20**(8), 823–856 (2006)

de La Beaujardière J (ed) (2003) OpenGIS Web Map Server Interface Implementation Specification Revision 1.1.1. OpenGIS Project Document 01-068r3 2003. http://portal.opengeospatial.org/files/?artifact_id=1081&version=1&format=pdf

de la Briandais, R.: File searching using variable length keys. In: Proceedings of the Western Joint Computer Conference, pp. 295–298 (1959)

Lacroix, Z., Boucelma, O., Essid, M.: A WFS-based mediation system for GIS interoperability. In: Workshop on Advances in Geographic Information Systems, pp. 23–28. ACM, New York (2002)

Ladd, A.M., Bekris, K.E., Marceau, G., Rudys, A., Wallach, D.S., Kavraki, L.E.: Using Wireless Ethernet for Localization. IROS, Lausanne (2002)

Ladd, A.M., Bekris, K.E., Rudys, A., Marceau, G., Kavraki, L.E., Wallach, D.S.: (2002). Robotics-based location sensing using wireless ethernet. In: 8th ACM MOBICOM, Atlanta, Georgia, September 23-28 2002

Laine, A.F.: Wavelets in temporal and spatial processing of biomedical images. Annu. Rev. Biomed. Eng. **02**, 511–550 (2000)

Lai, T.L.: Sequential changepoint detection in quality control and dynamical systems. J. R. Stat. Soc. B **57**, 613–658 (1995)

Lake, M.W.: The use of pedestrian modelling in archaeology, with an example from the study of cultural learning. Environ. Plann. B: Plann. Des. **28**, 385–403 (2001)

Lake, M.W., Woodman, P.E., Mithen, S.J.: Tailoring GIS software for archaeological applications: An example concerning viewshed analysis. J. Arch. Sci. **25**(1), 27–38 (1998)

Lakoff, G.: Women, Fire, and Dangerous Things: What Categories Reveal About the Mind, 614. The University of Chicago Press, Chicago (1987)

Lalonde W (ed) (2002) Styled Layer Descriptor Implementation Specification. OpenGIS Project Document 02-070. https://portal.opengeospatial.org/files/?artifact_id=1188

Lambert, L., Wolshon, B.: Analysis of Reverse Flow Traffic Operations, Phase I: Urban Sporting Event Measurement and Evaluation. Louisiana State University, Department of Civil and Environmental Engineering, Baton Rouge, LA (2002)

Lamy, S., Ruas, A., Demazeau, Y., Jackson, M., Mackaness, W.A., Weibel, R.: The Application of Agents in Automated Map Generalization. Paper presented at Proceedings of the 19th International Cartographic Conference, Ottawa 1999

Lance, K., Bassolé, A.: SDI and national information and communication infrastructure (NICI) integration in Africa. Inf. Technol. Dev. **12**, 333–338 (2006)

Landt, J.: Shrouds of Time: The History of RFID. http://www.aimglobal.org/technologies/rfid/resources/shrouds_of_time.pdf, Association for Automatic Identification and Mobility: Retreived 10/10/2006 (2001)

Land Trust Alliance: About LTA: Local Land Trust Success Stories. Retrieved December 23 (2006) from http://www.lta.org/

Langacker, R.W.: Foundations of Cognitive Grammar – Theoretical Prerequisites, 1. Stanford University Press, Stanford, CA (1991), 516 pp

Langford, M., Unwin, D.: Generating and mapping population density surfaces within a geographical information system. Cartogr J. **31**, 21–26 (1994)

Lang, L.: Managing Natural Resources with GIS. ESRI Press (1998)

Langran, G.: Time in Geographic Information Systems. PhD thesis, University of Washington (1999)

Langran, G.: Time in geographic information systems. Taylor and Francis, London (1992)

Laporte, G., Semet, F.: Classical heuristics for the capacitated VRP. In: Toth, P., Vigo, D. (eds.) The Vehicle Routing Problem, pp. 109–128. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia (2002)

Larson, R.R.: Geographical information retrieval and spatial browsing. In: Smith, L.C., Gluck, M. (eds.) Geographical Information Systems and Libraries: Patrons, Maps, and Spatial Information, pp. 81–124. Publications Office, Graduate School of Library & Information Science, University of Illinois at Urbana-Champaign, Champaign, IL, USA (1996)

Larsson, E.G.: Cramér-Rao bound analysis of distributed positioning in sensor networks. IEEE Signal Process. Lett. **11**(3), 334–337 (2004)

Laser-Scan. www.laser-scan.com and from December 2006, www.1spatial.com

Laube, P., Imfeld, S., Weibel, R.: Discovering relative motion patterns in groups of moving point objects. Int. J. Geogr. Inf. Sci. **19**, 639–668 (2005)

Laube, P., van Kreveld, M., Imfeld, S.: Finding REMO – detecting relative motion patterns in geospatial lifelines. In: Fisher, P.F. (ed.) Developments in Spatial Data Handling, Proceedings of the 11th International Symposium on Spatial Data Handling, pp. 201–214. Springer, Berlin Heidelberg (2004)

Laurini, R., Thompson, D.: Fundamentals of Spatial Information Systems. The APIC Series. Academic Press, London & San Diego, CA (1992)

La Vigne, N. G., Groff, E. R.: The Evolution of Crime Mapping in the United States: From the descriptive to the analytic. In: Hirschfield, A., Bowers, K. (eds.) Mapping and Analyzing Crime Data, pp. 203–221. University of Liverpool Press, Liverpool, UK (2001)

La Vigne, N., Wartell, J.: Crime Mapping Case Studies: Success in the Field vol. 1. Police Executive Research Forum and National Institute of Justice, US Department of Justice, Washington, DC (1998)

Lawder, J.K., King, P.J.H.: Querying multi-dimensional data indexed using the hilbert space filling curve. SIGMOD Record **30**(1), (2001)

Lawler, E.L., Lenstra, J. K., Rinnooy Khan, A.H.G., Shmoys, D. B.: The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization. John Wiley & Sons, Chichester (1985)

Lawrence, S., Giles, C.L., Tsoi, A.C., Back, A.D.: Face recognition: a convolutional neural network approach. IEEE Transactions on Neural Networks **8**(1), 1–27 (1997)

Lawson, A.B., Böhning, D., Lessafre, E., Biggeri, A., Viel, J.F., Bertollini, R. (eds.) Disease Mapping and Risk Assessment for Public Health. Wiley, Chichester (1999)

Lawson, A.B., Clark, A.: Markov chain Monte Carlo methods for clustering in case event and count data in spatial epidemiology. In: Halloran, M.E., Berry, D. (eds.) Statistics and Epidemiology: Environment and Clinical Trials, pp 193–218. Springer, New York (1999)

Lawson, A.B.: Cluster modelling of disease incidence via rjmcmc methods: a comparative evaluation. Stat. Med. **19**, 2361–2376 (2000)

Lawson, A.B., Denison, D.: Spatial cluster modelling: an overview. In: Lawson, A.B., Denison, D. (eds.) Spatial Cluster Modelling, chapter 1, pp. 1–19. CRC press, New York (2002)

Lawson, A.B.: On using spatial Gaussian priors to model heterogeneity in environmental epidemiology. The Statistician. **43**, 69–76 (1994). Proceedings of the Practical Bayesian Statistics Conference

Lawson, A.B.: Statistical Methods in Spatial Epidemiology, 2nd edn. Wiley, New York (2006)

Lazaridis, I., Mehrotra, S.: Approximate selection queries over imprecise data. In: ICDE '04: Proceedings of the 20th International Conference on Data Engineering, pp. 140–140. IEEE Computer Society, Washington, DC, USA (2004)

Lazaridis, I., Mehrotra, S.: Progressive approximate aggregate queries with a multi-resolution tree structure. In: SIGMOD '01: Proceedings of the 2001 ACM SIGMOD international conference on Management of data, pp. 401–412. ACM Press, New York, NY, USA (2001)

Lazos, L., Poovendran, R.: SeRLoc: secure range-independent localization for wireless sensor networks. In: WiSe '04: Proc. 2004 ACM workshop on Wireless security, pp. 21–30 (2004)

LeBeau, J.L.: Distance Analysis:Man With A Gun Calls For Servicein Charlotte, N.C., 1989. In: Levine, N. (ed.) CrimeStat III: A Spatial Statistics Program for the Analysis of Crime Incident Locations (version 3.0), Chapter 4. Ned Levine & Associates and the National Institute of Justice, Houston, TX, Washington, DC (2004)

LeBeau, J.L.: Four Case Studies Illustrating the Spatial-temporal Analysis of Serial Rapists. Police Stud. **15**, 124–145 (1992)

LeBeau, J.L.: Patterns of Stranger and Serial Rape Offending: Factors Distinguishing Apprehended and At Large Offenders. J. Crim. Law Criminol. **78**(2), 309–326 (1987)

Leberl, F., Gruber, M., Ponticelli, Bernoegger, S., Perko, R.: The Ultracam large format aerial digital camera system. Proceedings ASPRS 2003 annual conference, Anchorage, Alaska, May 2003, digitally on CD-Rom (2003) 7 pages

Lecordix, F., Lemarie, C.: Managing Generalization Updates in IGN Map Production. In: Mackaness, W.A., Ruas, A., Sarjakoski, L.T. (eds.) Generalization of Geographic Information: Cartographic Modeling and Applications. Elsevier, Oxford (2007)

Lecordix, F., Trévisan, J., Le Gallic J.M., Gondol, L.: Clarity Experimentations for cartographic generalisation in production. 9th ICA Workshop of the ICA Commission on Generalisation and Multiple Representation, 25 June 2006 – Portland, USA. Available at http://ica.ign.fr/

Lee, D.C., Bradshaw, G.A.: Making Monitoring Work for Managers: Thoughts on a conceptual framework for improved monitoring within broad-scale ecosystem management. (1998) http://icebmp.gov/spatial/lee_monitor/preface.html (Dec. 26, 2001)

Lee, D., Xu, J., Zheng, B.: Data management in location-dependent information services. IEEE Pervasive Computing, **1**(3), 2002

Lee, M.L., Hsu, W., Jensen, C.S., Cui, Teo, K.L.: Supporting Frequent Updates in R-Trees: A Bottom-Up Approach. In: Proceedings of the VLDB Conference, pp. 608–619. Berlin, Germany, 9–12 Sept 2003

LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Incognito: Efficient Full-Domain *k*-Anonymity. In: Proceedings of the ACM International Conference on Management of Data (SIGMOD), pp. 29–60. Baltimore, Maryland, USA (2005)

LeFevre, K., DeWitt, D., Ramakrishnan, R.: Mondrian Multidimensional *k*-Anonymity. In: Proceedings of the International Conference on Data Engineering (ICDE), Atlanta, GA, USA (2006)

Legleiter, C.J., Goodchild, M.F.: Alternative representations of in-stream habitat: classification using remote sensing, hydraulic modeling, and fuzzy logic. Int. J. Geogr. Inf. Sci. **19**(1), 29–50 (2005)

Lehmann, F., Cohn, A.G.: The EGG/YOLK reliability hierarchy: Semantic data integration using sorts with prototypes. In: Proc. Conf. on Information Knowledge Management, pp. 272–279. ACM Press, New York (1994)

Lehto, L., Sarjakoski, L.T.: Real-time generalization of XML-encoded spatial data for the web and mobile devices. Int. J. Geogr. Inf. Sci. **19**, 957–973 (2005)

Leidner, J.L.: Toponym Resolution in Text: Annotation, Evaluation and Applications of Spatial Grounding of Place Names. PhD thesis, School of Informatics, University of Edinburgh, Edinburgh, Scotland, UK (2007)

Leino, A., Mannila, H., Pitkänen, R.: Rule discovery and probabilistic modeling for onomastic data. In: Lavrac, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) Knowledge Discovery in Databases: PKDD 2003. Lecture Notes in Artificial Intelligence 2838, pp. 291–302. Springer, Heidelberg (2003)

Leitner, H.: The Politics of Scale and Networks of Spatial Connectivity: Transnational Interurban networks and the Rescaling of Political Governance in Europe. In: Sheppard, E., McMaster, B. (eds.) Scale and Geographic Inquiry: Nature Society and Method, pp. 236–255 Blackwell Publishing, Malden, MA (2004)

Lembo, J.: Lecture 9, Spatial Autocorrelation. http://www.css.cornell.edu/courses/620/lecture9.ppt#256,1 (2006). Accessed 10 Apr 2006

Lenntorp, B.: Paths in space-time environments: A time geographic study of movement possibilities of individuals. Lund Studies in Geography Number 44, Royal University of Lund, Sweden (1976)

Lenz, H.J., Jurgens, M.: The Ra*-tree: An improved r-tree with materialized data for supporting range queries on olap-data. In: DEXA Workshop 1998

LeSage, J., Pace, R.K.: Conditioning upon All the Data: Improved Prediction via Imputation, Journal of Real Estate Finance and Economics, vol. 29, pp. 233–254 (2004)

LeSage, J., Pace, R.K.: Introduction. In: LeSage, J.P. and Pace, R.K. (eds.) Advances in Econometrics: vol. 18: Spatial and Spatiotemporal Econometrics, pp. 1–30. Elsevier Ltd, Oxford (2004)

LeSage, J., Pace, R.K.: A Matrix Exponential Spatial Specification, Journal of Econometrics (2006)

LeSage, J.P.: The Theory and Practice of Spatial Econometrics. Department of Economics, University of Toledo (1999)

Leutenegger, S.T., Edgington, J.M., Lopez, M.A.: Str: A simple and efficient algorithm for r-tree packing. In: Proc. of International Conference on Data Engineering (ICDE), Birmingham, UK, 7–11 Apr 1997, pp. 497–506

Levene, M.: An Introduction to Search Engines and Web Navigation. Pearson, (2005) ISBN13: 9780321306777

Levenshtein, V.: Binary codes capable of correcting deletions, insertions, and reversals. In Soviet Physics – Doklady 10, 10, pp. 707–710 (1966)

Levine, N.: Bayesian Journey to Crime estimation (update chapter). In: Levine, N. (ed.) CrimeStat III: A Spatial Statistics Program for the Analysis of Crime Incident Locations (version 3.1). Ned Levine & Associates, Houston, TX and the National Institute of Justice, Washington, DC. March 2007. Available at http://www.icpsr.umich.edu/crimestat (2007)

Levine, N.: CrimeStat III: A Spatial Statistics Program for the Analysis of Crime Incident Locations (version 3.1). Ned Levine & Associates and the National Institute of Justice, Houston ,TX ,Washington, DC (2007)

Levine, N.: CrimeStat III Version 3.0, A Spatial Statistics Program for the Analysis of Crime Incident Locations (2005)

Levine, N.: Risk-adjusted nearest neighbor hierarchical clustering, In: Levine, N. (ed.) CrimeStat III: A Spatial Statistics Program for the Analysis of Crime Incident Locations (version 3.0), Chapter 6. Ned Levine & Associates and the National Institute of Justice, Houston, TX , Washington, DC (2004)

Levine, N.: Spatial statistics and GIS: software tools to quantify spatial patterns. Journal of the American Planning Association **62**(3), 381–392 (1996)

Levitin, A.: Introduction to The Design and Analysis of Algorithms. Addison Wesley (2003)

Lewin, K.: Field Theory in Social Science. Selected Theoretical Papers. D. Cartwright. Harper&Row, New York (1951)

Lewis, D.C.: September's Great Escape: New Information System Helps Manage Hurricane Evacuations. Roads Bridges **September**, 40–42 (2001)

Liao, H., Tim, U.S.: An interactive modeling environment for non-point source pollution control. J. Am. Water Resour. Assoc. **3**(3), 591–603 (1997)

Liao, S., Lopez, M.A., Leutenegger, S.T.: High dimensional similarity search with space-filling curves. In: Proc. of Intl. Conf. on Data Engineering, ICDE, Heidelberg, Germany, April 2001, pp. 615–622

Li, B.: Client-side Processing for Geographic Information Services. In: Geoinformatics for Global Change Studies and Sustainable Development, Manchu Li, et al. (eds.), Berkeley: CPGIS (2002)

Li, B.: A Component Perspective on Geographic Information Services, Cartography and Geographic Information Systems. **27**(1), 75–86 (2000)

Li, B., Zhang, L.: Distributed Spatial Catalog Service on the CORBA Object Bus. Geoinformatica. **4**(3), 253–269

Li, D.: Geospatial Semantic Web Research at LAITS. http://www.ncgia.ucsb.edu/projects/nga/docs/Di_Position.pdf

Lieberman, J.: Geospatial Semantic Web Interoperability Experiment Report.http://portal.opengeospatial.org/files/?artifact_id=15198 Accessed 22 March 2007

Lieberman, J., Lutz, M., Pehle, T., Kolas, D., Morris C., Dean M., Probst, F., Klien, E.: Geospatial Semantic Web: An Interoperability Experience. http://www.semantic-conference.com/Presentations_PDF/Lieberman-Joshua.pdf (2007). Accessed 31 Aug 2007

Lieberman, J., Pehle, T., Dean, M. (2005) Semantic Evolution of Geospatial Web Services. In: W3C Workshop on Frameworks for Semantic in Web Services. Available via. http://www.w3.org/2005/04/FSWS/Submissions/48/GSWS_Position_Paper.html. Accessed 9-10 June 2005

Li, F., Cheng, D., Hadjieleftheriou, M., Kollios, G., Teng, S.H.: On Trip Planning Queries in Spatial Databases. In: Proceedings of the 9th International Symposium on Spatial and Temporal Databases (SSTD'05), volume 3633 of Lecture Notes in Computer Science, pages 273–290. Springer, August (2005)

Li, F., Cheng, D., Hadjieleftheriou, M., Kollios, G., Teng., S.-H.: On trip planning queries in spatial databases. In: SSTD: Proc. of the

International Symposium on Advances in Spatial and Temporal Databases, pp. 273–290 (2005)

Li, F., Golden, B.L., Wasil, E.A.: Very large-scale vehicle routing: New test problems, algorithms, and results. In: Computers & Operations Research, Forthcoming (2004)

Li, J., Narayanan, R., Revesz, P.: A shape-based approach to change detection and information mining in remote sensing. In: Chen, C.H. (ed.) Front. Remote Sensing Inf. Process. 63–86. WSP (2003)

Li, J.Y.Y.; Han, J.: Continuous k-nearest neighbor search for moving objects. In: SSDBM '04: Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM'04), p. 123. IEEE Computer Society, Washington, DC, USA (2004)

Lilienfeld, D.E.: Foundations of Epidemiology, 3rd edn. Oxford University Press, Oxford (1994)

Li, L., Li, Y., Piltner, R.: A new shape function based spatiotemporal interpolation method. In: Proc. of the First International Symposium on Constraint Databases 2004, Lecture Notes in Computer Science, vol. 3074, 25–39. Springer, Berlin, Heidelberg, New York (2004)

Li, L., Revesz, P.: A Comparison of Spatio-temporal Interpolation Methods. In: Proc. of the Second International Conference on GIScience 2002. Lecture Notes in Computer Science, vol. 2478, pp. 145–160. Springer, Berlin, Heidelberg, New York (2002)

Li, L., Revesz, P.: Interpolation methods for spatio-temporal geographic data. J. Comput. Environ. Urban Syst. **28**(3), 201–227 (2004)

Li, L., Revesz, P.: The relationship among GIS-oriented spatiotemporal databases. In: Proc. of the Third National Conference on Digital Government Research, Boston (2003)

Li, L.: Spatiotemporal Interpolation Methods in GIS. Ph.D Thesis, University of Nebraska-Lincoln, Lincoln, Nebraska, May (2003)

Li, L., Zhang, X., Piltner, R.: A spatiotemporal database for ozone in the conterminous u.s. In: Proc. of the Thirteenth International Symposium on Temporal Representation and Reasoning, pp. 168–176. IEEE (2006)

Li, M., Sampigethaya, K., Huang, L., Poovendran, R.: Swing & swap: user-centric approaches towards maximizing location privacy. In: WPES '06: Proceedings of the 5th ACM workshop on Privacy in electronic society, Alexandria, Virginia, USA, pp. 19–28. ACM Press, New York, NY, USA (2006). doi: 10.1145/1179601.1179605

Lim, Y.Y.: Modeling and Evaluating Evacuation Contraflow Termination Point Designs. Master's Thesis, Department of Civil and Environmental Engineering, Louisiana State University (2003)

Lin, B., Su, J.: Handling frequent updates of moving objects. In: Proceedings of ACM CIKM, pp. 493–500. Bremen, Germany, 31 Oct–5 Nov 2005

Lin, C., Nevatia, R.: Building Detection and Description from Monocular Aerial Images, Proceeding DARPA Image Understanding Workshop, Palm Springs, CA, pp. 461–468 (1996)

Lindell, M.K., Bolton, P.A., Perry, R.W., Stoetzel, G.A., Martin, J.B., Flynn, C.B.: Planning Concepts and Decision Criteria for Sheltering and Evacuation in a Nuclear Power Plant Emergency. National Environmental Studies Project, Atomic Industrial Forum Inc., AIF/NESP-031 (1985)

Lindell, M.K., Prater, C.S.: Development of a decision support system for hurricane evacuation management, pp. 238–294. Iowa State University, Armes, Iowa (2002)

Lindenberger, J.: Test results of laser profiling for topographic terrain survey. In: Proceedings 42nd Photogrammetric Week, Stuttgart, 11–16 September 1989

Lindley, D.V., Smith, A.F.M.: Bayes estimates for the linear model. J. Royal Stat. Soc., Series B **34**, 1–41 (1972)

Lin, K-I., Jagadish, H.V., Faloutsos, C.: The TV-tree: An index structure for high-dimensional data. VLDB J. 3(4):517–542 (1994) [DBLP:journals/vldb/LinJF94]

Link, J.: Technical Coordinator: Northwest Forest Plan – the first 10 years (1994–2003): status and trends of northern spotted owl populations and habitat. General Technical Report PNW-GTR-648. Portland, OR. U. S. Department of Agricultures, Forest Service, Pacific Northwest Research Station. 176 p (2005)

Lin, X., Yuan, Y., Wang, W., Lu, H.: Stabbing the Sky: Efficient Skyline Computation over Sliding Windows. In: Proceedings of ICDE'05, IEEE Computer Society, pp. 502–513 Tokyo, Japan (2005)

Lipscomb, C., Farmer, M.: Household diversity and market segmentation within a single neighborhood. Ann. Reg. Sci. **39**(4), 791–810 (2005)

Li, Q., Moon, B.: Indexing and Querying XML Data for Regular Path Expressions. In: Proceedings of the 27th International Conference on Very Large Data Bases, Rome, Italy, pp. 361–370 (2001)

Lisboa Filho, J., Iochpe, C., Borges, K.A.: Analysis patterns for GIS data schema reuse on urban management applications. CLEI Electron. J. **5**, 1–15 (2002)

Lisboa Filho, J., Iochpe, C.: Specifying analysis patterns for geographic databases on the basis of a conceptual framework. In: ACM Symposium on Advances in Geographic Information Systems, pp. 9–13. Kansas City, 5–6 Nov 1999

Litman, T.: Lessons from katrina and rita: What major disasters can teach transportation planners. J. Trans. Eng. **132**(1), 11–18 (2006)

Liu C., Franzese, O., Bhaduri, B.: A Front-end Geographic Information System for an Emergency Evacuation Model. In: Proceedings of the Association of American Geographers, Chicago, IL (2006)

Liu C., Franzese, O., Bhaduri, B.: Oak Ridge Evacuation Modeling System (OREMS): A PC-based Computer Tool for Emergency Evacuation Planning. In: Proceedings of the GIS for Transportation Symposium, Columbus, OH (2006)

Liu, C.W.: Decision support system for managing ground water resources in the Choushui River alluvial in Taiwan. J. Am. Water Res. Assoc. **40**(2), 431–442 (2004)

Liu, C., Wu, K., He, T.: Sensor localization with ring overlapping based on comparison of received signal strength indicator. In: IEEE Mobile Ad-hoc and Sensor Systems (MASS), Oct. (2004)

Liu, H., Ban, J., Ma, W., Mirchandani, P.B.: Adaptive Control Based Real Time Traffic Management for Emergency Evacuation. ASCE J. Urban Plann. Dev. (in press 2007)

Liu, Y., Lai, X., Chang, G.L.: A Two-Level Integrated Optimization Model for Planning of Emergency Evacuation: A Case Study of Ocean City Under Hurricane Evacuation. Presented at The 84th Annual Meeting of Transportation Research Board, Washington, D. C. (2005)

Li, X: RSS-based location estimation with unknown pathloss model. IEEE Trans. Wireless Commun. **5**(12), 3626–3633 (2006)

Li, Z., Yan, H., Ai, T., Chen, J.: Automated building generalization based on urban morphology and Gestalt theory. Int. J. Geogr. Inf. Sci. **18**(5), 513–534 (2004)

Lloyd, S.P.: Least squares quantization in pcm. IEEE Transactions on Information Theory, **28**(2), 127–135 (1982)

Location Intelligence: The New Geography of Business, Business Week Research Services (2006)

Loeve, M.: Fonctions aleatoires de seconde ordre. Processus Stochastiques et Mouvement Brownien, Gauthier-Villars, Paris (1948)

Lomet, D., Salzberg, B.: Access methods for multiversion data. In: SIGMOD, pp. 315–324 (1989)

London Resilience Team: London Prepared (2005)

Longhorn, R.: Geospatial Standards, Interoperability, Metadata Semantics and Spatial Data Infrastructure, background paper for NIEeS Workshop on Activating Metadata, Cambridge, UK 6–7 July 2005. http://archive.niees.ac.uk/talks/activating_metadata/Standards_Overview_and_Semantics.pdf

Longley, P., Batty, M.: Spatial Analysis: Modeling in a GIS Environment. Wiley, Cambridge, UK (1996)

Longley, P.A., Goodchild, M.F., Maguire, D.J., Rhind, D.W.: Geographic Information Systems and Science. John Wiley & Sons, Chichester, West Sussex, England (2001)

Longley, P., Goodchild, M., Maguire, D., Rhind, D.: Geographic information systems and science, 2nd ed. John Wiley & Sons, Chichester, England (2005)

Longley, P., Goodchild, M., Maguire, D., Rhind, D.: Second Edition, Geographical Information Systems, Volume 1 Principles and Technical Issues, Volume 2 Management Issues and Applications. John Wiley & Sons Publishing, New York, NY (1999)

Lorentzos, N.A.: A Formal Extension of the Relational Model for the Representation and Manipulation of Generic Intervals. PhD thesis, Birbeck College, University of London (1988)

Los Alamos National Laboratory: http://transims.tsasa.lanl.gov/. Accessed 10 Sept 2007

Loshin: Enterprise Knowledge Management (The Data Quality Approach). Morgan Kaufmann, San Francisco (2001). ISBN 0-12-455840-2

Loukopoulos, T., Kalnis, P., Ahmad, I., Papadias, D.: active caching of on-line-analytical-processing queries in WWW proxies. In: Proceedings of the 2001 International Conference on Parallel Processing (ICPP), Valencia, Spain, 3–7 Sept 2001, pp. 419–426

Lovas, G.: Modeling and Simulation of Pedestrian Traffic Flow. Trans. Res. B. **28**(6), 429–443 (1994)

Lowe, J.A.: Geospatial Semantic Web. http://www.geospatial-online.com/geospatialsolutions/article/articleDetail.jsp?id=163883. Accessed 21 March 2007

Lucas, H.C.: Performance and the use of an information system. Manag. Sci. **21**(8), 908–919 (1979)

Lu, C.C., Dunham, J.G.: Highly efficient coding schemes for contour lines based on chain code representations. IEEE Transaction on Communication **39**(10), 1511–1514 (1991)

Lu, C., Chen, D., Kou, Y.: Detecting spatial outliers with multiple attributes. In: 15th IEEE International Conference on Tools with Artifcial Intelligence, p. 122 (2003)

Lu, C.-T., Chen, D., Kou, Y.: Algorithms for spatial outlier detection. In: Proceedings of the Third IEEE International Conference on Data Mining, p. 597–600, Melbourne, Florida, United States, 19–22 Nov 2003

Lu, C.-T., Liang, L.R.: Wavelet fuzzy classification for detecting and tracking region outliers in meteorological data. In: GIS '04: Proceedings of the 12th annual ACM international workshop on Geographic information systems, p. 258–265, Washington DC, USA, 12–13 Nov 2004

Lu, D., Mausel, P., Brondízios, E., Moran, E.: Change detection techniques. Int. J. Remote Sens. **25**(12), 2365–2407 (2004)

Luhmann, T. (ed.): Nahbereichsphotogrammetrie in der Praxis. Wichmann, Heidelberg (2002)

Luhmann, T., Robson, S., Kyle, S., Harley, I.: Close-Range Photogrammetry. Whittles, Caithness, UK (2006)

Lunetta, R.S., Elvidge, C.D.: Remote sensing change detection: environmental monitoring methods and applications. p. 318. Ann Arbor Press, Chelsea, USA, (1998)

Lu, Q., George, B., Shekhar, S.: Capacity Constrained Routing Algorithms for Evacuation Planning: A Summary of Results, Advances in Spatial and Temporal Databases. In: Proceeding of 9th International Symposium on Spatial and Temporal Databases (SSTD'05), Angra dos Reis, Brazil, 22–24 Aug 2005

Lu, W., Mannen, S., Sakamoto, M., Uchida, O., Doihara, T.: Integration of imageries in GIS for disaster prevention support system. In: Proceedings of the XXth ISPRS Congress, Istanbul, Turkey 2004

Lynch, J.P., Kiremidjian, A.S., Law, K.H., Kenny, T., Carryer, E.: Issues in wireless structural damage monitoring technologies. In: 3rd World Conference on Structural Control (WCSC), Como, Italy (2002)

Lynch, K.: The Image of the City. MIT Press, Cambridge, Massachusetts (1960)

Lynch, M.P., Saalfield, A.: Conflation: Automated Map Compilation – A Video Game Approach. In: Proc. Auto-Carto 7 ACSM/ASP, Falls Church, VA, 11 Mar 1985

MacEachren, A., Brewer, I., Cai, G., Chen, C.: Visually enabled Geocollaboration to support data exploration and decision-making. In: Proceedings of the 21st International Cartographic Conference, 10–16 August: Durban, South Africa (2003)

MacEachren, A.M., Brewer, I.: Developing a conceptual framework for visually-enabled geocollaboration. Int. J. Geogr. Inf. Sci. **18**, 1–34 (2004)

MacEachren, A.M., Cai, G., Brewer, I., Chen, J.: Supporting map-based geocollaboration through natural interfaces to large-screen displays. Cartographic Persp. **54**, 16–34 (2006)

MacEachren, A.M., Cai, G., McNeese, M., Sharma, R., Fuhrmann, S.: Geocollaborative crisis management: Designing technologies to meet real-world needs. In: Proceedings of the 7th annual national conference on Digital government research. Digital Government Research Center (2006)

MacEachren, A.M., Robinson, A., Hopper, S., Gardner, S., Murray, R., Gahegan, M., Hetzler, E.: Visualizing Geospatial Information Uncertainty: What We Know and What We Need to Know. Cartogr. Geogr. Inform. Syst., **32**(3), 139–160 (2005)

MacEachren, A.M.: Some Truth With Maps: A Primer On Design And Symbolization. Association of American Geographers, Washington, DC (1994)

MacEachren, A.M., Taylor, D.R.F., (eds.): Visualization in Modern Cartography. Pergamon, London (1994)

Mackaness, W.A.: Automated cartography in a bush of ghosts. Cartogr. Geogr. Inf. Sci. **33**, 245–256 (2006)

Mackaness, W.A., Beard, M.K.: Use of Graph Theory to Support Map Generalization. Cartography and Geographic Information Systems. **20** 210–221 (1993)

Mackaness, W., Ruas, A., Sarjakoski, L.T.: Generalisation of Geographic Information: Cartographic Modelling and applications. Elsevier, Burlington, MA, USA (2007)

MacKay, D.J.: Gaussian Processes: A Replacement for Supervised Neural Networks? In: Lecture Notes of Tutorial at Neural Information Processing Systems (NIPS'97) (1997)

Ma, C.: The use of the variogram in construction of stationary time series models. J. Appl. Probability 41, 1093–1103 (2004)

Macy, M., Willer, R.: From Factors to Actors: Computational Sociology and Agent-Based Modeling. Ann. Rev. Soc. **28**, 143–166 (2002)

Madani, M.: Photogrammetric applications. In: McGlone (ed.) Manual of Photogrammetry, 5th edn., pp. 1015–1104. American Society for Photogrammetry and Remote Sensing, Bethesda, Maryland (2004)>

Madden, S., Franklin, M.J., Hellerstein, J., Hong, W.: TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks. In: Proceedings of the 5th USENIX Symposium on Operating Systems Design and Implementation (OSDI), pp. 131–146. December 2002

Madden, S., Franklin, M.J., Hellerstein, J.M., Hong, W.: The Design of an Acquisitional Query Processor for Sensor Networks. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data (SIGMOD), pp. 491–502. San Diego, CA, June (2003)

Madhavan, J., Halevy, A.Y., Cohen, S., Dong, X.L., Jeffrey, S.R., Ko, D., Yu, C.: Structured data meets the web: a few observations. IEEE Data Eng. Bull. **29**, 19–26 (2006)

Madhavan, J., Halevy, A.Y.: Composing mappings among data sources. In: Proceedings of the Very Large Data Bases Conference (VLDB) 2003, Sept. 9–12, Berlin, Germany

MAF/Tiger Accuracy Improvement Program, http://www.census.gov/geo/mod/maftiger.html. This article has been prepared for information purposes only. It is not designed to constitute definitive advice on the topics covered and any reliance placed on the contents of this article is at the sole risk of the reader

Maguire, D.J., Longley, P.A.: The emergence of geoportals and their role in spatial data infrastructures. Comput. Environ. Urban Syst. **29**(1), 3–14 (2005)

Maguire, D.J.: Towards a GIS platform for spatial analysis and modeling. In: Maguire, D.J., Batty, M., Goodchild, M.F. (eds.) GIS, Spatial Analysis, and Modeling, pp. 19–39. ESRI Press, Redlands, California (2005)

Mahmassani, H.S., Hayssam, A.S., Zhou, X.: DYNASMART-P V1.2 User's Guide. University of Maryland at College Park, Austin (2006)

Mahmassani, H.S., Sbayti, H., Zhou, X.: DYNASMART-P, Intelligent Transportation Network Planning Tool, Version 1.0 User's Guide, Unversity of Maryland, August (2004)

Maidment, D.R., and Djokic, D.: Hydrologic and Hydraulic Modeling Support with Geographic Information Systems. ESRI Press, Redlands, CA (2000)

Maidment, D.R., Morehouse, S. (eds.): Arc Hydro: GIS for Water Resources, p. 218, ISBN 1-5894-8034-1. ESRI Press (2002)

Maidment, D.R., Robayo, O., Merwade, V.: Hydrologic modeling. In: Maguire, D.J., Batty, M., Goodchild, M.F. (eds.) GIS, Spatial Analysis, and Modeling, pp. 319–332. ESRI Press (2005)

Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D.: Wireless sensor networks for habitat monitoring In: 1st Int'l Workshop on Wireless Sensor Networks and Applications, Atlanta, Georgia 2002

Malczewski, J.: A GIS-based multicriteria decision analysis: A survey of the literature. Int J Geogr Inf Sci **20**(7), 703–726 (2006)

Malczewski, J.: GIS and multicriteria decision analysis. John Wiley & Sons, New York (1999)

Malerba, D., Esposito, F., Lisi, F.A.: Mining spatial association rules in census data. In: Proceedings of 4th International Seminar on New Techniques and Technologies for Statistics (NTTS 2001), Crete (2001)

Maling, D.H., Measurement from Maps: Principles and Methods of Cartometry. Pergamon, New York (1989)

Malinowski, E., Zimányi, E.: Advanced data warehouse design: From conventional to spatial and temporal applications. Springer, Berlin, Heidelberg (2007, in press)

Malinowski, E., Zimanyi, E.: Representing spatiality in a conceptual multidimensional model. In: Proceedings of the 12th annual ACM international workshop on geographic information systems, pp. 12–22. ACM Press, Washington DC (2003)

Mallat, S.: A Wavelet Tour of Signal Processing, 2nd edn. Academic, New York (1999)

Mamoulis, N., Cao, H., Kollios, G., Hadjieleftheriou, M., Tao, Y., Cheung, D.: Mining, indexing, and querying historical spatiotemporal data. In: Proceedings of the 10th ACM International Conference On Knowledge Discovery and Data Mining, pp. 236–245. ACM Press, New York (2004)

Mamoulis, N., Papadias, D.: Multiway spatial joins. ACM Trans. Database Syst. **26**(4), 424–475 (2001)

Mandelbrot, B.B.: Fractal Geometry of Nature. W.H. Freeman, New York (1977)

Mandell, R.B.: A guide to videokeratography. Int. Contact Lens Clin. **23**(6), 205–28 (1996)

Mani, I., Pustejovsky, J., Gaizauskas, R. (eds.): The Language of Time: A Reader. Oxford University Press, Oxford (2005)

Mannelli, A., Ferre, N., Marangon, S.: Analysis of the 1999 2000 highly pathogenic avian influenza (h7n1) epidemic in the main poultry-production area in northern Italy. Prev. Vet. Med., **73**, 273–285 (2006)

Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Mining Knowl Disc **1**(3), 241–258 (1997)

Mannila, H., Toivonen, H., Verkamo, A.I.: Discovering frequent episodes in sequences. In: First International Conference on Knowledge Discovery and Data Mining (KDD'95, August), pp. 210–215. AAAI Press Montreal, Canada (1995)

Manolopoulos, Y., Nanopoulos, A., Papadopoulos, A.N., Theodoridis, Y.: Rtrees: Theory and Applications. Springer, Germany (2005)

Manolopoulos, Y., Theodoridis, Y., Tsotras, V.: Chapter 4: Access methods for intervals. In: Advanced Database Indexing. Kluwer, 2000

Mansbridge, J.: Beyond Adversary Democracy. Basic Books, New York (1980)

MapGuide. Davecentral: Mapguide plug-in. http://www3.autodesk.com/

Mapquest, http://www.mapquest.com (2007)

MapServer. Mapsever home

MAPS AND MANAGEMENT: COMSTAT EVOLVES, Raymond Dussault, Editor, Government Technlogy, April (2000) http://www.govtech.net/magazine/crimetech/Apr00/CTEComstat.php

MapViewer. Xerox parc map viewer. http://mapweb.parc.xerox.com/map/

Marchand, P., Brisebois, A., Bedard, Y., Edwards, G.: Implementation and evaluation of a hypercube-based method for spatio-temporal exploration and analysis. J. Int. Soc. Photogrammetry Remote Sensing **59**, 6–20 (2003)

Marian, A., Bruno, N., Gravano, L.: Evaluating top-k queries over web-accessible databases. ACM Transactions on Database Systems **29**(2), 319–362 (2004)

Marinoni, O.: A discussion on the computational limitations of outranking methods for land-use suitability assessment. Int J Geogr Inf Sci **20**(1), 69–87 (2006)

Mark, D.M., Egenhofer, M.J.: Geospatial lifelines. In: Gunther, O., Sellis, T., Theodoulidis, B. (eds.). Integrating Spatial and Temporal Databases. Dagstuhl-Seminar-Report No. 228, http://timelab.co.umist.ac.uk/events/dag98/report.html, Schloss Dagstuhl, Germany (1998)

Maroti, M., Kusy, B., Balogh, G., Volgyesi, P., Molnar, K., Nadas, A., Dora, S., Ledeczi, A.: Radio interferometric positioning. Institute for Software Integrated Systems, Vanderbilt University, Tech. Rep. ISIS-05–602, Nov. (2005)

Marsolo, K., Parthasarathy, S., Twa, M.D., Bullimore, M.A.: A model-based approach to visualizing classification decisions for patient diagnosis. In: Proceedings of the Conference on Artificial Intelligence in Medicine (AIME), Aberdeen, 23–27 July 2005

Marsolo, K., Twa, M., Bullimore, M.A., Parthasarathy, S.: Spatial modeling and classification of corneal shape. IEEE Transactions on Information Technology in Biomedicine (2006)

Martin, P.H., et al.: Development of a GIS-based spill management information system. J. Hazard. Mat. **112**(3), 239–252 (2004)

Mas, J.-F.: Monitoring land-cover changes: a comparison of change detection techniques. Int. J. Remote Sens. **20**(1), 139–152 (1999)

Mason, B., Dragicevic, S.: Web GIS and knowledge management systems: An integrated design for collaborative community planning. In: Balram, S., Dragicevic, S. (eds.) Collaborative Geographic Information Systems, pp. 263–284. Idea Group Inc. (2006)

Masser, I.: The future of spatial data infrastructures. ISPRS Workshop on Service and Application of Spatial Data Infrastructure, XXXVI (4/W6). Swets & Zeitlinger B.V, Hangzhou, China (2005)

Mather, P.M.: Computer Processing of Remotely-Sensed Images, 3rd edn. p. 442. John Wiley and Sons, Hoboken, NY (2004)

Matheus, A.: Declaration and enforcement of fine-grained access restrictions for a service-based geospatial data infrastructure. In: Proceedings of the10th ACM Symposium on Access Control Models and Technologies, pp. 21–28. ACM, New York (2005)

MATLAB: The Student Edition of MATLAB, The Language of Technical Computing, Version 5, User's Guide. The Mathworks Inc., New Jersey (1997)

Matousek, J.: Computing dominance in E^n. Information processing letters, Vol. 38 pp. 277–278 (1991)

Matousek, J.: Efficient Partition Trees. Discret. Comput. Geom. **8**, 432–448 (1992)

Matouśek, J.: Efficient partition trees. Discrete Comput. Geom. **8**(3), 315–334 (1992)

Matousek, J.: Geometric range searching. ACM Computing Survey, **26**(4), 1994

Matsuoka, R., Sone, M., Fukue, K., Cho, K., Shimoda, H.: Quantitative analysis of image quality of lossy compression images. http://www.isprs.org/istanbul2004/comm3/papers/348.pdf

Maturana, H.R.: Biologie der Realität. Suhrkamp, Frankfurt am Main (1998)

Maune, D. (ed.): Digital Elevation Model Technologies and Applications: The DEM Users Manual, 2nd edn. American Society for Photogrammetry and Remote Sensing, Bethesda (2006)

Mauve, M., Widmer, A., Hartenstein, H.: A survey on position-based routing in mobile ad hoc networks, IEEE Network, vol. 15, no. 6, pp. 30–39 (2001)

Mauvière, E., Coulomb, I.: Geoclip, Interactive Flash Map Gallery, http://www.geoclip.fr/an/p24_atlas.php, (2006) accessed 2006-12-28

Ma, X., Shekhar, S., Xiong, H., Zhang, P.: Exploiting a page-level upper bound for multi-type nearest neighbor queries. In: ACM GIS, pp. 179–186 (2006)

Mayer, L., Barbor, K., Boudreau, P., Chance, T., Fletcher, C., Greening, H., Li, R., Mason, C., Metcalf, K., Snow-Cotter, S., Wright, D.: A Geospatial Framework for the Coastal Zone: National Needs for Coastal Mapping and Charting. National Academies, Washington, DC (2004)

May, J.M.: Medical Geography: Its Methods and Objectives. Geogr. Rev. **40**, 9–41 (1950)

Ma, Y., Soatto, S., Kosecka, J., Sastry, S.S.: An Invitation to 3-D Vision. Springer Verlag, New York (2004)

Mazzoleni, P., Bertino, E., Crispo, B., Sivasubramanian, S.:XACML policy integration algorithms:~not to be confused with XACML policy combination algorithms! In: Proceedings of 11th ACM Symposium on Access Control Models and Technologies, pp. 219–227. ACM, New York (2006)

McColl, C., Aggett, G.: Boom or Bust? A Case Study of PSS & PPGIS Implemented within a Rural Setting. In: Proceedings of the Annual PPGIS Conference of the Urban and Regional Information Systems Association (URISA), pp. 145–157. University of Wisconsin, Madison, WI (2004)

McColl, C., Aggett, G.: Land use forecasting and hydrologic model integration for improved land use decision support. J. Env. Manag. **84**, 494–512 (2007)

McCormick, B., DeFanti, T.A., Brown, M.D.: Visualization in Scientific Computing. Comput. Graphics. **21**, (1987)

McCrady, D.D., Doyle, L., Forstrom, H., Dempsy, T., Martorana, M.: Mobile ranging with low accuracy clocks. IEEE Trans. Microwave Theory Tech. **48**(6), 951–957 (2000)

McCurley, K.S.: Geospatial Mapping and Navigation of the Web. In: Proceedings of the 10th International Conference on the World Wide Web, Hong Kong, 1–5 May 2001

McDermott, D.: A temporal logic for reasoning about processes and plans. Cogn. Sci. **6**, 101–55 (1982)

McGeough, J.: Location-based services and topology. In: GITA (2001)

McGlone, J.C. (ed.): Manual of Photogrammety, 5th edn. American Society for Photogrammetry and Remote Sensing. Bethesda, Maryland (2004)

McHarg, I.: Design with Nature. American Museum of Natural History (1969)

McIntosh, J., Yuan, M.: Assessing similarity of geographic processes and events. Trans. GIS **9**, 223–245 (2005)

McKelvin, M.L., Williams, M.L., Berry, N.M.: Integrated radio frequency identification and wireless sensor network architecture for automated inventory management and tracking applications. In: Proceedings of the Richard Tapia Celebration of Diversity in Computing Conference (TAPIA'05), Albuquerque, New Mexico, USA, 19–22 Oct 2005

McKeown, D.M., Pane, J.F.: Alignment and Connection of Fragmented Linear Feature in Aerial Imagery. In: Proceedings of 1985 IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, pp. 55–61 (1985)

McLeod K.S.: Our sense of Snow: the myth of John Snow in medical geography. Soc. Sci. & Medicine **50**, 923–935 (2000)

McMaster, B.R., Shea, K.S.: Generalization in Digital Cartography. Association of American Geographers, Washington, D.C. (1992)

McMaster, R., Buttenfield, B.: Map Generalization. Longman, Harlow (1991)

McPherson, T., Brown, M.: Estimating daytime and nighttime population distributions in U.S. cities for emergency response activities. In: Proceedings of the Symposium on Planning, Nowcasting, and Forecasting in the Urban Zone, 84th AMS Annual Meeting, Seattle WA, 11–15 Jan 2004

Meagher, D.J: Octree 3-D Geospatial Modeling Application: Artillery Shadows. http://www.octree.com/ArtilleryShadow/report.htm. Accessed 12 March 2007

Megalooikonomou, V., Davataikos, C., Herskovits, E. H.: Mining lesion-deficit associations in a brain image database. KDD, San Diego, CA, USA (1999)

Meguerdichian, S., Koushanfar, F., Potkonjak, M., Srivastava, M.B.: Coverage problems in wireless ad-hoc sensor networks. In: INFOCOM, pp. 1380–1387 (2001)

Melesse, Assefa M., Graham, Wendy D., Jordan, Jonathan D.: Spatially Distributed Watershed Mapping and Modeling: GIS-Based Storm Runoff Response and Hydrograph Analysis: Part 2. J. Spat. Hydro. **3** (2003)

Mellor-Crummey, J.M., Whalley, D.B., Kennedy, K.: Improving memory hierarchy performance for irregular applications. In: Proc. of the Intl. Conf. on Supercomputing, ICS, Rhodes, Greece, June 1999, pp. 425–433

Melnikov, H.: Open solution for location based services in WLAN environment. Master thesis, Tampere University of Technology (2004)

Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity flooding: a versitile graph matching algorithm and its application to schema matching. In: Proceedings of the International Conference on Data Engineering 2002, June 3–6, Madison, WI

Melnik, S., Rahm, E., Bernstein, P.A.: Rondo: A programming platform for generic model management. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD) 2003, June 9–12, San Diego, CA

Meng, L., Zipf, A., Reichenbacher, T. (eds.): Map-based Mobile Services, Theories, Methods and Implementations Springer, Berlin Heidelberg, New York (2005)

Mennecke, B.E., Crossland, M.D., Killingsworth, B.L.: Is a map more than a picture? The role of SDSS technology, subject characteristics, and problem complexity on map reading and problem solving. MIS Quart. **24**(4), 601–629 (2000)

Mennis, J.: Generating surface models of population using dasymetric mapping. Prof. Geogr. **55**, 31–42 (2003)

Mercier, F., Baujard, O.: Voronoi diagrams to model forest dynamics in french guiana. In: 2rd International Conference on GeoComputation University of Otago, (1997)

Merleau-Ponty, M.: Phänomenologie der Wahrnehmung. Walter De Gruyter & Co., Berlin (1966)

Merlo, S., Colin, A.: Developing a multidimensional GIS framework for archaeological excavations. In: CIPA 2005, XX International Symposium Torino, Torino (2005)

Mesarovic, M.D., Macko, D., Takahara, Y.: Theory of hierarchical, multilevel, systems. Mathematics in science and engineering. Academic Press, New York (1970)

Metadata Education Project, Wyoming Geographic Information Science Center. http://www.sdvc.uwyo.edu/metadata/education.html (2006)

Metadata Reference Guide for Federal Geographic Data Committee Metadata, Metadata Advisory Group, Massachusetts Institute of Technology Libraries. http://libraries.mit.edu/guides/subjects/metadata/standards/fgdc.html (2004)

Metropolitan Washington Council of Governments. 2004 performance of regional high-occupancy vehicle facilities on freeways in the washington region. Analysis of Person and Vehicle Volumes (2005)

Meyer, R., Meydenbauer, A.: Baukunst in historischen Photographien. Fotokinoverlag, Leipzig (1985) ISBN: 3731100630, 259 pages

Meyer, S.C.: Analysis of base flow trends in urban streams, northeastern Illinois, USA. Hydrogeol. J. **13**: 871–885 (2005)

Meyers, L., Newman, M.E.J., Martin, M., Schrag, S.: Applying network theory to epidemics: Control measures for outbreaks of mycoplasma pneumonia. Emerging Infectious Diseases**9**(2), 204–210 (2003)

Meyerson, A., Williams, R.: On the Complexity of Optimal *K*-Anonymity. In: Proceedings of the ACM Symposium on Principles of Database Systems (PODS), pp.223–228. Paris, France (2004)

Michel, S., Triantafillou, P., Weikum, G.: Klee: a framework for distributed top-k query algorithms. In VLDB 05: Proceedings of the 31st international conference on Very large data bases, pp. 637–648, Trondheim, Norway, 30 Aug–2 Sept 2005

Michon, P.E., Denis, M.: When and Why Are Visual Landmarks Used in Giving Directions? In: Montello, D.R. (ed.) Conference on Spatial Information Theory: Foundations of Geographic Information Science, pp. 292–305. Springer-Verlag, Berlin (2001)

Miehle, P., Livesley, S.J., C.S. Li, Feikema, P.M., Adams, M.A., Arndt, S.K.: Quantifying uncertainty from large-scale model predictions of forest carbon dynamics. Global Change Biol. **12**, 1421–1434 (2006)

Mihaila, G., Raschid, L., Vidal, M.E.: Querying, "Quality of Data" Metadata. In: Proceedings of the Third IEEE Meta-Data Conference, Bethesda, Maryland, April 1999

Mikhail, E.M., Bethel, J., McGlone, J.C.: Introduction to Modern Photogrammetry, John Wiley and Sons, New York (2001)

Millennium Problems, The Clay Mathematics Institute of Cambridge, Massachusetts. http://www.claymath.org/millennium/

Miller, H.J., Han, J.: Geographic data mining and knowledge discovery: An overview. In: Miller, H.J., Han, J. (eds.) Geographic Data Mining and Knowledge Discovery, pp. 3–32. Taylor and Francis, London (2001)

Miller, H.J.: Measuring space-time accessibility benefits within transportation networks: Basic theory and computational methods. Geogr. Anal. **31**, 187–212 (1999)

Miller, H.J.: Necessary space-time conditions for human interaction. Environ. Plann. B: Plann. Des. **32**, 381–401 (2005)

Miller, H.: Modeling accessiblity using space-time prism concepts within a GIS. Int. J.~Geogr. Inf. Syst. **5**, 287–301 (1991)

Miller, J., Mukerji, J. (2001): Model Driven Architecture (MDA). OMG. http://www.omg.org/cgi-bin/apps/doc?ormsc/01-07-01.pdf. date cited: September 2004

Miller, J., Mukerji, J. (2003): MDA Guide Version 1.0.1. OMG. http://www.omg.org/cgi-bin/doc?omg/03-06-01

Miller, J., Mukerji, J.: MDA Guide Version 1.0. In: Miller, J., Mukerji, J. (eds.), OMG Document: omg/2003-05-01. http://www.omg.org/mda/mda_files/MDA_Guide_Version1-0.pdf (2003)

Miller, R.C., Guertin D.P, Heilman, P.: Information technology in watershed management decision making. J. Am. Water Resour. Assoc. **40**(2), 349–357 (2004)

Miller, R.J., Haas, L.M., Hernández, M.A.: Schema mapping as query discovery. In: Proceedings of the Very Large Data Bases Conference (VLDB) 2000, Sept. 10–14, Cairo, Eqypt

Miller, S.: Photogrammetric Products. In: McGlone, J.C. (ed.) Manual of Photogrammetry, 5th edn. American Society for Photogrammetry and Remote Sensing, Bethesda (2004)

Milller, H.J.: A measurement theory for time geography. Geogr. Anal. **37**, 17–45 (2005)

Minsk, L.D.: Snow and Ice Control Manual for Transportation Facilities. (1998)

Miralles, A.: Ingénierie des modèles pour les applications environnementales. Ph.D. Thesis, Université des Sciences et Techniques du Languedoc, p. 338. Département d'informatique, Montpellier, France (2006)

Missaoui, R., Sarifuddin, M., Vaillancourt, J.: Similarity measures for efficient content-based image retrieval. In: IEE Proceedings – Visual Image Signal Process. 152: 6, December 2005

Mitasova, H., Mitas, L., Brown, W.M., Gerdes, D.P., Kosinovsky, I., Baker, T.: Modelling spatially and temporally distributed phenomena: new methods and tools for grass GIS. Int. J. Geogr. Inf. Syst. **9**(4), 433–446 (1995)

Mitchell, A.: The ESRI Guide to GIS Analysis. Vol. 2: Spatial Measurements and Statistics. ESRI, Redlands, CA, USA (2005)

Mitchell, J., Brennan, C., Card, D.: Real-time image-space outlining for non-photorealistic rendering. ACM SIGGRAPH 2002 Sketch (2002)

Mitchell, J., Reece, C.F.: Leveraging GIS to Manage Weather-Related Evacuations. Presented at The 14th Annual Meeting of ITS America, San Antonio, Oct 2004

Mitchell, T.M.: Machine Learning, 1st edn. McGraw-Hill, New York (1997)

Moellering, H. (ed.): World Spatial Metadata Standards. Elsevier, Oxford. (2005)

Moellering, H., Hogan, R.: Spatial Database Transfer Standards 2: Characteristics for Assessing Standards and Full Descriptions of the National and International Standards in the World. International Cartographic Association, Pergamon, Oxford (1996)

Moens, M., Steedman, M.: Temporal ontology and temporal reference. Comput. Linguist. **14**, 15–28 (1988)

Moglen, G.E., Gabriel, S.A., Faria, J.A.: A Framework for Quantitative Smart Growth in Land Development. J. Am. Water Resour. Assoc. **39**(4), 947–959 (2003)

Mokbel, M.F., Aref, W.G., El-Bassyouni, K., Kamel, I.: Scalable Multimedia Disk Scheduling. In: Proceedings of the International Conference on Data Engineering, ICDE, Boston, MA, March 2004, pp. 498–509

Mokbel, M.F., Aref, W.G.: Irregularity in multi-dimensional space-filling curves with applications in multimedia databases. In: Proc. of the 2nd Intl. Conf. on Information and knowledge Management, CIKM, Atlanta, GA, November 2001, pp. 512–519

Mokbel, M.F., Aref, W.G., Kamel, I.: Analysis of Multi-dimensional Space-Filling Curves. GeoInformatica **7**(3), 179–209 (2003)

Mokbel, M.F., Aref, W.G., Kamel, I.: Performance of Multi-Dimensional Space-Filling Curves. In: Proceedings of the ACM Symposium on Advances in Geographic Information Systems, ACM GIS, McLean, VA, November 2002. pp. 149–154

Mokbel, M.F., Aref, W.A., Hambrusch, S.E., Prabhakar, S.: Towards scalable location-aware services: Requirements and research issues. In: GIS (2003)

Mokbel, M.F., Chow, C.Y., Aref, W.G.: The New Casper: Query Processing for Location Services without Compromising Privacy. In: Proceedings of the International Conference on Very Large Data Bases (VLDB), pp. 763–774. Seoul, Korea (2006)

Mokbel, M.F., Ghanem, T.M., Aref, W.G.: Spatio-temporal access methods. IEEE Data Eng. Bull. **26**(2), 40–49 (2003)

Mokbel, M.F., Ghanem, T.M., Aref, W.G.: Spatio-temporal access methods. Technical report, Department of Computer Sciences, Purdue University

Mokbel, M., Xiong, X., Aref, W.: SINA: Scalable Incremental Processing of Continuous Queries in Spatio-temporal Databases. ACM SIGMOD, Paris, France 13–18, June 2004

Molenaar, M.: An Introduction to the Theory of Spatial Object Modeling for GIS. Taylor and Francis, London (1998)

Möller, J.: Spatial Statistics and Computational Method. Springer, New York (2003)

Mollié, A.: Bayesian and empirical bayes approaches to disease mapping. In: Lawson, A., Biggeri, A., Boehning, D., Lesaffre, E., Viel, J.-F., Bertollini, R. (eds.) Disease Mapping and Risk Assessment for Public Health, chapter 2, pp. 15–29. Wiley, Chichester (1999)

Molnar, D., Wagner, D.: Privacy and security in library RFID: issues, practices, and architectures, In: CCS '04: Proceedings of the 11th ACM conference on Computer and communications security, pp.210–219. ACM Press, New York (2004)

Monestiez, P., Dubroca, L., Bonnin, E., Durbec, J.P., Guinet, C.: Geostatistical modelling of spatial distribution of Balenoptera physalus in the northwestern Mediterranean Sea from sparse count data and heterogeneous observation efforts. Ecol. Model. **193**, 615–628 (2006)

Money, A.T.D., Wegner, T.: The quantification of decision support within the context of value analysis. MIS Quart. **12**(2), 223–236 (1988)

Monge, P., Contractor, N.: Theories of Communication Networks. Oxford University Press, New York (2003)

Monmonier, M.: Geographic brushing: enhancing exploratory analysis of the scatterplot matrix. Geogr. Anal. **21**, 81–84 (1989)

Monmonier, M.: How to Lie with Maps. University of Chicago Press, Chicago (1991)

Monmonier, M.: Summary graphics for integrated visualization in dynamic cartography. Cartogr. Geogr. Inf. Syst.19, 23–36 (1992)

Montello, D.R., Goodchild, M.F., Gottsegen, J., Fohl, P.: Where's downtown?: Behavioural methods for determining referents of vague spatial queries. Spatial Cogni. Comput. **3**, 185–204 (2003)

Montello, D.: Scale and Multiple Psychologies of Space. In: Frank, A.U., Campari, I. (eds.) Spatial Information Theory: Theoretical Basis for GIS, p. 312–321. Springer-Verlag, Heidelberg-Berlin (1993)

Moon, B., Jagadish, H.V., Faloutsos, C., Salz, J.: Analysis of the clustering properties of hilbert space-filling curve. IEEE Trans. Knowl. Data Eng., TKDE **13**(1), 124–141 (2001)

Moore, D.A., Carpenter, T.E.: Spatial analytical methods and geographic information systems: Use in health research and Epidemiology. Epidemiol. Rev. **21**(2), 143–161 (1999)

Moore, E.H.: On certain crinkly curves. Trans. Am. Math. Soc. 72–90 (1900)

Moore, I.D., Grayson, R.B., Ladson, A.R.: Digital Terrian Modeling: A Review of Hydrological, Geomorphological and Biological Applications. J. Hydrol. Process **5**, 3–30 (1991)

Moore, I.D.: Hydrologic modeling and GIS. In: Goodchild, M.F., Steyaert, L.T., Parks, B.O., Johnston, C., Maidment, D.R., Crane, M.P., Glendinning, S. (eds.) GIS and Environmental Modeling: Progress and Research Issues, pp. 143–148. GIS World Books, Fort Collins, CO (1996)

Moran, P.A.P.: The interpretation of statistical maps. J. Roy. Stat. Soc. B **10**, 243–251 (1948)

Moran, P.A.P.: Notes on continuous stochastic phenomena. Biometrika **37**, 17–23 (1950)

Morehouse, S.: The ARC/INFO Geographic Information System. Computers Geosci. **18**(4), 435–443 (1992)

Morehouse, S.: ARC/INFO: a georelational model for spatial information. In: Proceedings of AUTOCARTO 8, pp. 388–97. ASPRS, Falls Church, Virginia (1985)

Morimoto, Y.: Mining frequent neighboring class sets in spatial databases. In: Int Proc. 7th ACM SIGKDD Conference on Knowledge and Discovery and Data Mining, pp. 353–358. ACM Press, San Francisco, California (2001)

Morrison, J., Wortman, K.: Special Issue. Implementing the Spatial Data Transfer Standard. Cartogr. Geogr. Inf. Syst. **19**(5), entire issue (1992)

Morville, P.: Ambient Findability: What We Find Changes Who We Become. O'Reilly Media, Sebastopol, CA (2005)

Moses, R.L., Krishnamurthy, D., Patterson, R.: A self-localization method for wireless sensor networks. EURASIP J. App. Sig. Proc. **4**, 348–358 (2003)

Mostafa, M.: Design and performance of the DSS. Fritsch (ed.) Photogrammetric Week 2003, Wichmann Verlag, Heidelberg, Germany, pp. 77–87, 2003. Also published in Proceedings International Workshop on Theory, Technology and Realities of Inertial/GPS Sensor Orientation 22.–23. September 2003, Castelldefels, Spain, digitally on CD-Rom (2003)

Mostafa, M., Hutton, J., Reid, B.: GPS/IMU products – the Applanix approach. Fritsch/Spiller (eds.) Photogrammetric Week 2001, pp. 63–83, Wichmann Verlag, Heidelberg, Germany (2001)

Mostashari, F., Kulldorff, M., Hartman, J., Miller, J., Kulasekera, V.: Dead bird clusters as an early warning system for West Nile Virus activity. Emerging Infect. Dis. **9**, 641–646 (2003)

Motani, M., Srinivasan, V., Nuggehalli, P.: PeopleNet: engineering a wireless virtual social network, International Conference Mobile Computing and Networking (MobiCom'05), Aug 2005

Motta, G., Rizzo, F., Storer, J. A.: Digital image compression. In: Ralston, A., Reilly, E.D., Hemmendinger, D. (eds.) The Encyclopedia of Computer Science, 4th edn. Grove Dictionaries (2000)

Mountain, D., MacFarlane, A.: Geographic information retrieval in a mobile environment: evaluating the needs of mobile individuals. J. Inf. Sci. (2007, in press)

Mouratidis, K., Yiu, M.L., Papadias, D., Mamoulis, N.: Continuous Nearest Neighbor Monitoring in Road Networks. VLDB (2006)

Moxon, B.: Defining Data Mining: The Hows and Whys of Data Mining, and How it Differs from Other Analytical Techniques. DBMS Data Warehouse Supplement, Miller Freeman, Inc., San Francisco, CA (1996)

Mukai, T., Kuriyama, S.: Geostatistical Motion Interpolation. ACM Trans. Graph. **24**(3), 1062–1070 (2005)

Müller, J.C., Lagrange, J.P., Weibel, R.: GIS and Generalization: Methodology and Practice. In: Masser, I., Salge, F. (eds.) GISDATA 1. Taylor and Francis, London (1995)

Müller, M., Poth, A.: deegree–components and framework for creation of SDIs. Geoinformatics **7** 2004

Munro, R., Chawla, S., Sun, P.: Complex spatial relationships. In: Proceedings of the 3rd IEEE International Conference on Data Mining, ICDM 2003, pp. 227–234. IEEE Computer Society (2003)

Murase, H., Nayar, S.K.: Three-dimensional object recognition from appearance-parametric eigenspace method. Systems and Computers in Japan **26**(8), 45–54 (1995)

Murray, C., Blowney, J., Xie, J., Xu, T., Yuditskaya, S., Zhang, Z.: Oracle Spatial GeoRaster, 10*g* Release 2 (10.2). Oracle, Redwood Shores, CA, USA (2005)

Murray, C.: Oracle Spatial User Guide and Reference, Release 9.2., Oracle Corporation (2002)

Murray, P.M., Mahmassani, H.S.: Model of Household Trip Chain Sequencing in an Emergency Evacuation. Transportation Research Board (CD-ROM), 82nd Annual Meeting, Washington, D.C (2003)

Mustafa, N., Krishnan, S., Varadhan, G., Venkatasubramanian, S.: Dynamic simplification and visualization of large maps. Int. J. Geogr. Inf. Sci. **20**, 273–302 (2006)

Muthukrishnan, K., Lijding, M., Havinga, P.: (2005). Towards smart surroundings: enabling techniques and technologies for localization. In: First International Workshop on Location- and Context-Awareness (LoCA 2005), Oberpfaffenhofen 2005, Faculty of Computer Science, University of Twente, Enschede, The Netherlands

Myles, G., Friday, A., Davies, N.: Preserving Privacy in Environments with Location-Based Applications. IEEE Pervasive Computing **2**(1):56–64 January–March (2003)

Myung, Y., Lee, C., Tcha, D.: On the Generalized Minimum Spanning Tree Problem. Networks (1995)

Naaman, M., Song, Y.J., Paepcke A., Garcia-Molina H.: Assigning Textual Names to Sets of Geographic Coordinates. J. Comput. Environ. Urban Syst. J. **30**(4):418–435 (2006)

Nabney, I.T.: Netlab: Algorithms for Pattern Recognition. Springer-Verlag, New York (2002)

Nachtigall, K.: Time depending shortest-path problems with applications to railway networks. Eur J Oper Res **83**:154–166 (1995)

Nagy, G., Wagle, S.: Geographic Data Processing. ACM Computing Surveys **11**(2), 139–181 (1979)

Naiburg, E.J., Maksimchuk, R.A.: UML for Database Design. Addison-Wesley, p. 300 (2001)

Nanopoulos, A., Vassilakopoulos, M., Manolopoulos, Y.: Performance evaluation of lazy deletion methods in r-trees. GeoInformatica **7**(4):337–354 (2003)

Nardi, B. (ed.): Context and consciousness – activity theory and human-computer interaction. The MIT Press, Cambridge, MA (1996)

NASA (1999) Proposal for a National Spatial Data Infrastructure Standards Project–FGDC Content Standard for Digital Geospatial Metadata: Extensions for Remote Sensing. http://www.fgdc.gov/standards/projects/FGDC-standards-projects/csdgm_rs_ex/standards/projects/csdgm_rs_ex/csdgm_rs_ex_pro.pdf. Accessed 9 Sept 2007

Nascimento M., Silva, J.: Towards historical R-trees. In: SAC, pp. 235–240 (1998)

National Academies of Science. Priorities for GEOINT Research at the National Geospatial-Intelligence Agency. (2006) ISBN: 0–309-65862–4

National Academy of Sciences: Remote Sensing with Special Reference to Agriculture and Forestry. p. 424. National Academy of Sciences, Washington, DC (1970)

National Aeronautics and Space Administration Geospatial Interoperability Office: Geospatial Interoperability Return on Investment Study, Greenbelt, MD, USA, April (2005). http://gio.gsfc.nasa.gov/docs/ROI%20Study.pdf

National Archives and Records Administration: Acceptable Archive Formats for Digital GIS Records, http://www.archives.gov/records-mgmt/initiatives/digital-geospatial-data-records.html, viewed on November 26 (2006)

National Consortium for Rural Geospatial Innovations: Mission, Goals, and Publications at www.ruralgis.org

National Geospatial-Intelligence Agency (NGA) in the United States of America. http://www.nga.mil/

National Information Standards Organization: Understanding Metadata. NISO Press, Bethesda, MD (2004). ISBN:

1-880124-62-9. http://www.niso.org/standards/resources/UnderstandingMetadata.pdf
National Oceanic and Atmospheric Administration. El Nino Web Page. www.elnino.noaa.gov/
National Research Council: Climate Data Records from Environmental Satellites: Interim Report, NOAA Operational Satellites. National Research Council (2004)
National Research Council: Making the Nation Safer: The Role of Science and Technology in Countering Terrorism. National Academy Press, Washington DC (2002)
National Science Foundation (1993) NSF Blue Ribbon Panel on High Performance Computing, October 19, 1993 (NSB 93–205).Available via http://www.nsf.gov/pubs/stis1993/nsb93205/nsb93205.txt. Accessed 30 September 2005
Natwichai J., Li, X.: Knowledge maintenance on data streams with concept drifting: International Symposium on Computation and Information Sciences (CIS), Shanghai, China, 2004, pp. 705–710
Naus, J.: The distribution of the size of the maximum cluster of points on the line. J. Am. Stat. Assoc. **60**, 532–538 (1965)
Nauta, T.A., Bongco, A.E., Santos-Borja, A.C.: Set-up of a decision support system to support sustainable development of the Laguna de Bay, Philippines. Mar. Poll. Bull. **47**(1–6), 211–219 (2003)
Næsset, E., Gobakken, T., Holmgren, J., Hyyppä, H., Hyyppä, J., Maltamo, M., Nilsson, M., Olsson, H., Persson, Å., Söderman, U.: Laser scanning of forest resources: the Nordic experience. Scand. J. Forest Res. **19**, 482–499 (2004)
NCGIA (1997) Interoperating GISs. Report of a Specialist Meeting Held under the Auspices of the Varenius Project. December 5–6, 1997, Santa Barbara, California. http://ncgia.ucsb.edu/conf/interop97/interop_toc.html. Accessed 11 Sept 2007
NCHRP: A Generic Data Model for Linear Referencing Systems, Transportation Research Board, Washington, DC (1997)
Neal, R.M.: Bayesian Learning for Neural Networks. Lecture Notes in Statistics No. 118. Springer-Verlag, New York (1996)
Neal, R.M.: Monte Carlo Implementations of Gaussian Process Models for Bayesian Regression and Classification. Technical Report 9702, Department of Statistics, University of Toronto, January 1997
Nebert, D.D. (2004) Developing Spatial Data Infrastructures: The SDI Cookbook, Version 2.0. http://www.gsdi.org/pubs/cookbook/cookbookV2.0.pdf. Accessed 12 Oct 2005
Nebert D (ed) (2002) OpenGIS Catalog Service Specification. OpenGIS Project Document 04-021r3. http://portal.opengeospatial.org/files/?artifact_id=5929&version=2
Neill, D.B., Moore, A.W., Sabhnani, M., Daniel, K.: Detection of emerging space-time clusters. In: Proceedings of SIGKDD 2005, pp. 218–227 (2005)
Nelson, R.C., Samet, H.: A Consistent Hierarchical Representation for Vector Data. In: Proceedings of the ACM SIGGRAPH Conference, pp. 197–206. Dallas Texas, USA, 18–22 Aug 1986
Neteler, M., Grasso, D., Michelazzi, I., Miori, L., Merler, S., Furlanello, C.: An integrated toolbox for image registration, fusion and classification. Int. J. Geoinf. Special Issue on FOSS/GRASS 2004 & GIS-IDEAS 2004, **1**(1), 51–60 (2005)
Neteler, M., Mitasova, H.: Open Source GIS: A GRASS GIS Approach, 3rd edn. Springer, NY (2007)
Neteler, M., Mitasova, H.: Open Source GIS: A GRASS GIS Approach. Number 773 in SECS. Kluwer Academic Publishers/Springer, Boston, first edition, June 2001. ISBN: 1–4020-8064-6; also published as eBook: ISBN 1-4020-8065-4
Neuffer, D., Hopewell, T., Woodsford, P.: Integration of Agent-based Generalisation with Mainstream Technologies and other System Components. ICA Workshop on Generalisation and Multiple Representation 20–21 August 2004 Leicester, UK Also available at http://www.laser-scan.com/papers/Neuffer-v2-ICAWorkshop.pdf
Neumann, A.: Dynamic Loading of Vector Geodata for SVG Mapping Applications Using Postgis, PHP and getURL()/XMLHttpRequest() (2005), http://www.carto.net/papers/svg/postgis_geturl_xmlhttprequest/, accessed 2006–11–12
Neumann, A.: SVG GUI (2003–2006), http://www.carto.net/papers/svg/gui/, accessed 2006–11–12
Neumann, A.: Use of SVG and ECMAScript Technology for E-Learning Purposes, in: Proceedings of the ISPRS Workshop Commissions VI/1 VI/2, Tools and Techniques for E-Learning, Potsdam, Germany, June 1–3 (2005)
Neumann, A., Winter, A.M.: Carto.net SVG page: articles, examples and tutorials (2000–2006), http://www.carto.net/papers/svg/, accessed 2006–11–12
Neumann, A., Winter, A.M.: Webmapping with Scalable Vector Graphics (SVG): Delivering the promise of high quality and interactive web maps, In: Peterson, M. (ed.) Maps and the Internet, p. 197–220. Elsevier, Kidlington, Oxford, UK (2003)
Newell, C.J., Rifai, H.S., Bedient, P.B.: Characterization of Non-Point Sources and Loadings to Galveston Bay, p. 33. Galveston Bay National Estuary Program (GBNEP-15), Clear Lake, Houston (1992)
Newman, M.E.J.: The structure and function of complex networks. SIAM Rev. **45**, 167–256 (2003)
Newsweek: An Internet of Things. http://www.msnbc.msn.com/id/3068871/, Newsweek: Retreived 10/10/2006 (2006)
NGA (National Geospatial-Intelligence Agency): Geospatial Intelligence (GEOINT) Basic Doctrine: Publication 1–0 (2006) Online at http://www.nga.mil/NGASiteContent/StaticFiles/OCR/geo_pub1.pdf
Ng, V., Kamada, T.: Concurrent Accesses to R-Trees. In: Proceedings of Symposium on Advances in Spatial Databases, Singapore, 23–25 June 1993
Niblack, W., Barber, R.: The QBIC project: querying images by content using color, texture and shape. In: Proc. SPIE Storage and Retrieval for Image and Video Databases, February 1994
Niculescu, D., Nath, B.: Ad hoc positioning system. In: IEEE Globecom 2001, vol. 5, pp. 2926–2931 (2001)
Niculescu, D.: Positioning in ad-hoc sensor networks. IEEE Network, 18(4), 24–29
Niedermeier, R., Reinhardt, K., Sanders, P.: Towards optimal locality in mesh-indexing. In: Proc. of the 11th Intl. Symp. on Fundamentals of Computation Theory, FCT, Krakow, Poland, September 1997, pp. 364–375
Niedzwiadek, H.: Where is the Value in Location Services, LBS360.NET. (http://www.lbs360.net/LBSArticles/HN.Where%27s%20the%20Value%20in%20Location%20Services.pdf), July 1 (2002)
Nielsen, J.: Usability Engineering. Morgan Kaufmann, New York (1993)
Nielson, J., Panda, S.S.: Predictive modeling and mapping of fasting blood glucse: A preliminary study of diabetes among Hispanics in SE Idaho. Presented in the Intermountain GIS User's Conference, Helena, Montana, USA, 4–8 April 2006
Niemann, B., Limp, F.: Evaluation of community decision support software. National Consortium for Rural Geospatial Innovations in America. University of Wisconsin, Madison, WI. www.ruralgis.org, see Publications (2004)

Nienhaus, M., Döllner, J.: Edge-enhancement – an algorithm for real-time non-photorealistic rendering. Journal of WSCG, 11(2):346–353 (2003)

Nieto, M.: Public video surveillance: Is it an effective crime prevention tool? Technical report, California Research Bureau Report, June 1997

Nievergelt, J., Hinterberger, H., Sevcik, K.C.: The grid file: An adaptable, symmetric multikey file structure. ACM Trans Database Syst **9**(1):38–71 (1984)

Nisha de Silva, F.: Challenges in Designing Spatial Decision Support Systems for Evacuation Planning. Natural Hazards Research Working Paper, (2000)

Nittel, S., Duckham, M., Kulik, L.: Information Dissemination in Mobile Ad-hoc Geosensor Networks. In: Proceedings of the 3rd International Conference on Geographic Information Science (GIScience 2004), Springer LNCS, College Park, Maryland 2004

Nittel, S., Labrinidis, A., Stefanidis, A.: Advanced in Geosensor Networks. Springer LNCS (in press 2007)

Nittel, S., Trigoni, N., Ferentinos, D., Neville, F., Nural, A., Pettigrew, N.: A drift-tolerant model for data management in ocean sensor networks In: Proceedings of MobiDE'07, in conjunction with SIGMOD, Beijing, China 2007

An, N., Kothuri, R., Ravada, S.: Improving performance with bulk inserts in Oracle R-trees. In: Proceedings of the International Conference on Very Large Data Bases (VLDB), Berlin, Germany, 9–12 Sept 2003

Nolan, J.: An agent-based architecture for distributed imagery and geospatial computing PhD thesis, George Mason University (2003)

Nord, G., Jabon, D., Nord J.: The Global Positioning System and the Implicit Function Theorem. SIAM Rev. **40**(3), 692–696 (1998)

Norman, D.: The Design of Everyday Things, 257. Doubleday, New York (1988)

Norsys Software Corp.: Netica Bayesian Belief Network Software. Acquired from http://www.norsys.com/ (2006)

North, D.C.: Institutions, Institutional Change and Economic Performance. Cambridge University Press, Cambridge, UK (1990)

Northrup, J.D., Markosian L.: Artistic silhouettes: A hybrid approach. In: Proceedings of NPAR (2000) 2000

Nothegger, C., Winter, S., Raubal, M.: Selection of salient features for route directions. Spatial Cogni. Comput. **4**, 113–136 (2004)

NVO (2006) National Virtual Observatory. http://www.us-vo.org/grid.cfm. Accessed 28 Sept 2006

Nyerges, T., Jankowski, P., Tuthill, D., Ramsey, K.: Collaborative water resource decision support: Results of a field experiment. Ann. Association Am. Geogr. **96**, 699–725 (2006)

Nyerges, T.L.: Coupling GIS and spatial analytic models. In: Breshanan, P., Corwin, E., Cowen, D. (eds.) Proceedings of the 5th International Symposium on Spatial Data Handling. pp. 534–543. Charleston, SC. Humanities and Social Sciences Computing Laboratory, University of South Carolina (1992)

Nyerges, T.L., Montejano, R., Oshiro, C., Dadswell, M.: Group-based geographic information systems for transportation improvement site selection. Transp. Res. Part C-Emerging Technol. **5**, 349–369 (1997)

Oak Ridge National Laboratory: LandScan Global Population Project. Oak Ridge National Laboratory, Tennessee. http://www.ornl.gov/sci/landscan/ (2007). Accessed 17 Sept 2007

OASIS. ebXML Registry Information Model (ebRIM). http://www.oasis-open.org/committees/regrep/documents/2.0/specs/ebrim.pdf. Accessed 13 Sept 2007

OASIS, Reference Model for Service Oriented Architecture, version 1.0. http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf (2006)

OASIS SOA Reference Model Technical Committee: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm

Öberg, S.: Hägerstrand and the remaking of Sweden. Prog. Hum. Geogr. **29**, 340–349 (2005)

Object Management Group.: OMG Unified Modeling Language Specification, Version 1.5. Object Management Group (2003)

Object Management Group.: OMG Unified Modeling Language Specification, Version 1.5. Object Management Group, Needham, MA (2003)

Object Management Group.: Unified Modeling Language: Superstructure, Version 2.0,.Object Management Group (2005)

Object Management Group.: Unified Modeling Language: Superstructure, Version 2.0,.Object Management Group, Needham, MA (2005)

O'Callaghan, L., Mishra, N., Meyerson, A., Guha, S., Motwani, R.: Streaming-data algorithms for high-quality clustering. In: Proceedings of IEEE International Conference on Data Engineering, San Jose, California, 2002, pp. 685–694

Ochola, W.O., Kerkides, P.: An integrated indicator-based spatial decision support system for land quality assessment in Kenya. Comp. Electron. Agric. **45**(1–3), 3–26 (2004)

O'Connor, M.C.: Gen 2 EPC Protocol Approved as ISO 18000–6C. http://www.rfidjournal.com/article/articleview/2481/1/1/, The RFID Journal: Retreived 11/05/2006 (2006)

O'Connor, M.C.: RFID takes attendance—and heat. RFID J. **February 16** (2005)

OGC 00-014r1, Guidelines for Successful OGC Interface Specifications

OGC 02-006, OpenGIS Abstract Specification Topic 12: OpenGIS Service Architecture

OGC 03-002r9 (2006) Binary Extensible Markup Language (BXML) Encoding Specification, version 0.0.8, Craig Bruce, http://www.opengeospatial.org/standards/bp. Accessed 20 Aug 2006

OGC 03-025, OpenGIS Web Services Architecture

OGC 05-008, OGC Web Services Common Specification, version 1.0.0

OGC 05-010, URNs of definitions in ogc namespace

OGC 05-50 (2005) GML Performance Investigation by CubeWerx, version 1.0.0, Craig Bruce, http://www.opengeospatial.org/standards/gml. Accessed 20 Aug 2006

OGC (2000), Topic 15:Imagery Exploitation Services. https://portal.opengeospatial.org/files/?artifact_id=7199. Accessed 29 Apr 2007

OGC (2000), Topic 16: Image Coordinate Transformation Services. http://portal.opengeospatial.org/files/?artifact_id=7200. Accessed 29 Apr 2007

OGC 2001, Introduction to OGC Web Services

OGC (2004), Topic 7: Earth Imagery. https://portal.opengeospatial.org/files/?artifact_id=7467. Accessed 29 Apr 2007

OGC (ed.) (2005): OpenGIS® Implementation Specification for Geographic Information – Part 1: Common architecture (Version 1.1.0)

OGC (ed.): OpenGIS Simple Features Specification for SQL (Revision 1.1). OGC (1999)

OGC. EO Products Extension Package for ebRIM (ISO/TS 15000–3) Profile of CSW 2.0. OGC 06–131. http://portal.opengeospatial.org/files/?artifact_id=17689. Accessed 13 Sept 2007

OGC: OGC Abstract Specification, Topic 6 – The coverage type and its subtypes, Version 4. Wayland, Massachusetts (1999)

OGC. OGC Adopts ebRIM for Catalogues. http://www.opengeospatial.org/pressroom/pressreleases/655. Accessed 13 Sept 2007

OGC. OGC™ Catalogue Services Specification, OGC 04–021r3. http://portal.opengeospatial.org/files/?artifact_5929&Version=2 Accessed 13 Sept 2007

OGC, Open Geospatial Consortium, Open Web Services, Phase 4 (2006), Topology Quality Assessment Service in Geoprocessing Workflow thread at http://www.opengeospatial.org/projects/initiatives/ows-4

OGC. OpenGIS® Catalogue Services Specification 2.0 – ISO19115/ISO19119 Application Profile for CSW 2.0. OGC 04–038r2. https://portal.opengeospatial.org/files/?artifact_id=8305. Accessed 13 Sept 2007

OGC, OpenGIS Location Services (OpenLS): Core Services, OGC 05-016. http://portal.opengeospatial.org/files/?artifact_id=8836 (2005)

OGC website: http://www.opengeospatial.org/

Ogden, F.L., Garbrecht, J., DeBarry, P.A. Johnson, L.E.: GIS and distributed watershed models II: Modules, interfaces, and models. J. Hydrol. Eng. **6**(6), 515–523 (2001)

OGP Surveying & Positioning Committee (2006) EPSG Geodetic Parameter Dataset v 6.11.2. http://www.epsg.org. Accessed 1 Jan 2007

OGR (2007): OGR Vector Formats (2007) http://www.gdal.org/ogr/ogr_formats.html. Accessed 22 Feb 2007

Oh, S., Russell, S., Sastry, S.: Markov chain Monte Carlo data association for general multiple-target tracking problems. In: Proceedings of the 43rd IEEE Conference on Decision and Control, Paradise Island, Bahamas (2004)

Okabe, A., Boots, B., Sugihara, K., Chiu, S.N.: Spatial Tessellations, Concepts and Applications of Voronoi Diagrams, 2nd edn. John Wiley and Sons Ltd., (2000)

O'Keefe, J., Nadel, L.: The Hippocampus as a Cognitive Map. Clarendon Press, Oxford (1978)

Okunieff, P., Siegel, D., Miao, Q.: Location Referencing Methods for Intelligent Transportation Systems (ITS) User Services: Recommended Apagesoach, In: Proc. GIS-T Conference, pp. 57–75 (1995)

Olea, R.: Optimum Mapping Techniques. Kansas Geological Survey, Lawrence, Kansas (1975)

Oliveira, R., Bernardo, L., Pinto, P.: Flooding techniques for resource discovery on high mobility MANETs, Workshop on Wireless ad hoc Networks (2005)

Olivera, F.: Extracting Hydrologic Information from Spatial Data for HMS Modeling. ASCE J. Hydrol. Eng. **6**(6), 524–530 (2001)

Olivera, F., Maidment, D.: GIS-Based Spatially Distributed Model for Runoff Routing. Water Resources Research **35**(4), 1155–1164 (1999)

Oliver, M.A., Webster, R., Lajaunie, C., Muir, K.R., Parkes, S.E., Cameron, A.H., Stevens, M.C.G., Mann, J.R.: Binomial cokriging for estimating and mapping the risk of childhood cancer. IMA J. Math. Appl. Med. Biol. **15**, 279–297 (1998)

OMB Circular A-16 (2002) http://www.whitehouse.gov/omb/circulars/a016/a016_rev.html. Accessed 10 Sept 2007

OMG – Object Management Group (2007) Unified Modeling Language. Available at http://www.uml.org

OMG (Object Management Group), CORBATM/IIOPTM Specification 3.0.3. http://www.omg.org/technology/documents/formal/corba_iiop.htm

Onsrud, H., et al.: Public Commons of Geographic Data: Research and Development Challenges. Geographic Information Science, Third International Conference, GIScience, Adelphi, MD, USA, 2004. Springer, Heidelberg (2004)

Onsrud, H.J., Johnson J.P., Lopez, X.: Protecting personal privacy in using geographic information systems. Photogramm. Eng. Image Process. **60**:1083–1095 (1994)

Onsrud, H., Poore, B., Rugg, R., Taupier, R., Wiggins, L.: The future of the spatial information infrastructure. In: McMaster, R.B., Usery, E.L. (eds.) A Research Agenda for Geographic Information Science, pp. 225–255. CRC, Boca Raton (2004)

Ooi, B.C., McDonell, K.J., Sacks-Davis, R.: Spatial kd-tree: An indexing mechanism for spatial databases. In: Proceedings of the IEEE COMPSAC Conference, Tokyo, pp. 433–438 (1987)

Open Geospatial Consortium, Inc.: "About OGC". http://www.opengeospatial.org/ogc (2007)

Open Geospatial Consortium, Inc. Filter Encoding Implementation Specification, OGC Website: http://www.opengeospatial.org/ (2001)

Open Geospatial Consortium, Inc. Vision and Mission, OGC Website: http://www.opengeospatial.org/about/?page=vision (2006)

OpenGeoSpatial Consortium (OGC) specification on Web Feature Service (http://www.opengeospatial.org/standards/wfs): According to the OGC website, (http://www.opengeospatial.org), OGC is a non-profit, international, voluntary consensus standards organization that is leading the development of standards for geospatial and location based services. It provides technical documents that detail agreed-upon interfaces or encodings which when implemented by software developers will ensure that the resulting components work together coherently

Open Geospatial Consortium: Opengis specifications (standards). http://www.opengeospatial.org/standards

The Open Geospatial Consortium White Papers at www.opengeospatial.org/pressroom/papers

OpenGIS Consortium (2006) OpenGIS Web Map Service Implementation Specification.http://www.opengeospatial.org/standards/wms 23 Mar 2007

OpenGIS: The OpenGIS Guide—Introduction to Interoperable Geoprocessing and the OpenGIS Specification. Open GIS Consortium, Wayland, MA (1996)

Openshaw, S.: A mark 1 geographical analysis machine for the automated analysis of point data sets. Int. J. GIS **1**(4):335–358 (1987)

OpenStreetMap Community, OpenStreetMap, http://www.openstreetmap.org/ (2006) accessed 2007-01-05

Oppermann, R. (ed.): Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software. Lawrence Erlbaum Associates, Hillsdale, NJ (1994)

Oppermann, R., Specht, M.: Adaptive mobile museum guide for information and learning on demand, in human-computer interaction. In: Bullinger, H.J., Ziegler, J. (eds.). Proceedings of HCI International '99, vol. 2: Communication, Cooperation, And Application Design, pp. 642–646. Erlbaum, Mahwah (1999)

Oracle. (2006) 10g Release 2 Spatial User's Guide and Reference. http://www.oracle.com/products/spatial. Accesed 2006

Oracle: Oracle Data Warehousing. Available at www.oracle.com (2006)

Orda, A., Rom, R.: Minimum Weight Paths in Time-Dependent Networks. In: Networks: An International Journal **21** (1991)

Ord, J.K., Getis, A.: Local Spatial Autocorrelation Statistics: Distributional Issues and an Application. Geogr. Anal. **27**, 286–306 (1995)

Ord, J.K., Rees, M.: Spatial Processes: Recent Developments with Applications to Hydrology. In: Lloyd, E.H., et. al (eds.) The

Mathematics of Hydrology and Water Resources. Academic Press, New York (1979)

Ordnance Survey Great Britain (GB). Meeting the Master Challenge. 18 Sept 2001. Available at: http://www.ordnancesurvey.co.uk/oswebsite/media/news/2001/sept/masterchallenge.html

Ordnance Survey, Positional Accuracy Improvement website, www.ordnancesurvey.co.uk/PAI

Ordonez, C., Omiecinski, E.: Discovering association rules based on image content. In: Proceedings of the IEEE Advances in Digital Libraries Conference (ADL'99) (1999)

O'Reilly, T. (2005) What is Web 2.0. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html. Accessed 2 November 2004

Orenstein, J.A.: Redundancy in spatial databases. SIGMOD Conference pp. 295–305 (1989)

Orenstein, J.A., Spatial query processing in an object-oriented database system, Proceedings of the 1986 ACM SIGMOD international conference on Management data, p. 326–336, May 28–30, Washington, D.C., United States (1986)

Ormeling, F. Atlas information systems, In: Proceedings of the 17th International Cartographic Conference of the ICA, Barcelona, vol. 2, p. 2127–2133 (1995)

Ormeling, F.: Functionality of electronic school atlases. In: Köbben, B., Ormeling, F., Trainor, T. (eds.) Seminar on Electronic Atlases II, pp. 33–39. ICA Commission on National and Regional Atlases, Prague (1997)

Ormsby, T., Napoleon, E., Burke, R., Feaster, L., Groessl, C.: Getting to Know ArcGIS Desktop Second Edition: Basics of ArcView, ArcEditor, and ArcInfo. ESRI Press, Redlands (2004)

ORNL: Oak Ridge Evacuation Modeling System (OREMS) User' Guide. Oak Ridge National Laboratory, Oak Ridge (1999)

O'Rourke, J.: Computational Geometry in C. Cambridge University Press, New York (1988)

O'Rourke, J., Supowit, K.J.: Some NP-hard polygon decomposition problems. IEEE Transactions on Information Theory **29**(2), 181–190 (1983)

The Orton Family Foundation: CommunityViz, Version 1.3. Course Training Manual. http://www.communityviz.com (2002)

Ortuzar, J.D., Willumsen, L.G.: Modeling Transport, 3rd edn. John Wiley & Sons, New York (2001)

Orłowska, E. (ed.): Incomplete Information – Rough Set Analysis. Studies in Fuzziness and Soft Computing, vol. 13. Physica-Verlag, Heidelberg (1998)

Osman, I.H.: Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Ann. Oper. Res. **41**, 421–451 (1993)

Ostfeld, R., Glass, G., Keesing, F.: Spatial epidemiology: an emerging (or re-emerging) discipline. Trends Ecol. Evol. **20**, 328–336 (2005)

O'Sullivan, D.: Geogrpahical information science: time changes everything. Prog. Hum. Geogr. **29**, 749–756 (2005)

O'Sullivan, D., Morrison, A., Shearer, J.: Using desktop GIS for the investigation of accessibility by public transport: An isochrone approach. Int. J. Geogr. Inf. Sci. **14**, 85–104 (2000)

O'Sullivan, D., Unwin, D.J.: Geographic Information Analysis. Wiley, Hoboken, NJ, USA (2003)

Ou, C.-W., Gunwani, M., Ranka, S.: Architecture-independent locality-improving transformations of computational graphs embedded in k-dimensions. In: Proc. of the 9th ACM Intl. Conf. on Supercomputing, ICS, Barcelona, Spain, July 1995, pp. 289–298

Overmars, M.H.: The Design of Dynamic Data Strucutures. vol. 156 of LNCS. Springer-Verlag, Heidelberg (1983)

OWL Overview http://www.w3.org/TR/owl-features/

Oyvind, V.: Geographic Information Retrieval: An Overview. Internal Doctoral Conference, IDI, NTNU, Norway (2004)

Pace, R.K., Barry, R.: Quick Computation of Spatial Autoregressive Estimators, Geographical Analysis, vol. 29, pp. 232–246 (1997)

Pace, R.K., Barry, R., Sirmans, C.F.: Spatial Statistics and Real Estate. J. Real Estate Finance Econ. **17**, 5–13 (1998)

Pace, R.K., LeSage, J.: Spatial Autoregressive Local Estimation. In: Mur, J., Zoller, H., Getis A. (eds.) Recent Advances in Spatial Econometrics, pp. 31–51. Palgrave Publishers (2004)

Paciorek, C.J., Schervish, M.J.: Spatial modelling using a new class of nonstationary covariance functions. Environ. **17**, 483–506 (2006)

Paelinck, Jean H.P., Klaassen, L.H.: Spatial Econometric. Saxon House, England (1979)

Pagel, B.-U., Six, H.-W., Toben, H., Widmayer, P.: Towards an analysis of range query performance in spatial data structures. In: PODS, pp. 214–221 (1993)

Pahlavan, K., Krishnamurthy, P., Beneat, J.: Wideband radio propagation modeling for indoor geolocation applications. IEEE Comm. Magazine **36**, 60–65 (1998)

Pahlavan, K., Krishnamurthy, P., Hatami, A., Ylianttila, M., Mäkelä, J., Pichna, R., Vallström, J.: Handoff in hybrid mobile data networks. IEEE Personal Commun. Mag. **7**, 34–47 (2000)

Pahlavan, K., Li, X., Makela, J.: Indoor geolocation science and technology. IEEE Commun. Mag. **40**(2), 112–118 (2002)

Painho, M., Bacao, F.: Using Genetic Algorithms in Clustering Problems. GeoComputation 2000 (2000)

Pajarola, R., Widmayer, P.: An image compression method for spatial search. IEEE Trans. Image Process. **9**, (2000)

Pajarola, R., Widmayer, P.: Spatial indexing into compressed raster images: how to answer range queries without decompression. http://www.ifi.unizh.ch/vmml/admin/upload/MMDBMS96.pdf

Palacio, A.: Legal Empowerment of the Poor: An Action Agenda for the World Bank. Available via http://rru.worldbank.org/Documents/PSDForum/2006/background/legal_empowerment_of_poor.pdf. Accessed 14 Aug 2007

Pal, A., Graettinger, A.J., Triche, M.H.: Emergency Evacuation Modeling Based On Geographical Information System Data. Transportation Research Board (CD-ROM) 82nd Annual Meeting, Washington, D.C. (2003)

Pallaro, P. (ed.): Authentic Movement. Jessica Kingsley Publishers, London (1999)

Pallottino, S., Scutellà, M.G.: Shortest Path Algorithms in Transportation Models: Classical and Innovative Aspects. In: Marcotte, P., Nguyen, S. (ed.) Equilibrium and Advanced Transportation Modelling, pp. 245–281. Kluwer Academic Publishers, New York (1998)

Palm, R.: Spatial Segmentation of the Urban Housing Market. Econom. Geogr. **54**(3), 210–21 (1978)

Panda, S.: Data mining application in production management of crop. Ph.D. Dissertation, North Dakota State University, Fargo, North Dakota, USA (2003)

Pandya, A. S., Macy, R. B.: Pattern Recognition with Neural Networks in C++. CRC Press, Boca Raton, FL. (1996)

Pan, F., Wang, B., Ren, D., Hu, X., Perrizo, W.: Proximal support vector machine for spatial data using peano trees: In: ISCA Computer Applications in Industry and Engineering, Las Vegas, Nevada, 2003, pp. 292–297

Pan, J., Kwok, J., Yang, Q., Chen, Y.: Multidimensional vector regression for accurate and low-cost location estimation in pervasive computing. IEEE Trans. on Knowledge and Data Engineering **18**(9), 1181–1193 (2006)

Pantazis, D., Donnay, J.P.: La conception SIG: méthode et formalisme, p. 343. Hermès, Paris (1996)

Papadias, D., Kalnis, P., Zhang, J., Tao, Y.: Efficient OLAP operations in spatial data warehouses. In: SSTD '01: Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases, pp. 443–459. Springer-Verlag, London, UK (2001)

Papadias, D., Tao, Y., Fu, G., Seeger, B.: An optimal and progressive algorithm for Skyline Queries. In: ACM SIGMOD, San Diego, CA, USA (2003)

Papadias, D., Tao, Y., Fu, G., Seeger, B.: Progressive Skyline Computation in Database Systems. ACM Trans. Database Syst. **30**(1),41–82 (2005)

Papadias, D., Zhang, J., Mamoulis, N., Tao, Y.: Query Processing in Spatial Network Databases, In: Proc. International Conference on Very Large Data Bases, pp. 802–813 (2003)

Papadimitriou, C., Steiglitz, K.: Some complexity results for the Traveling Salesman Problem. Proceedings of the eighth annual ACM symposium on Theory of computing, May (1976)

Papadopouli, M., Schulzrinne, H.: Design and implementation of a P2P data dissemination and prefetching tool for mobile users, First NY Metro Area Networking Workshop, IBM TJ Watson Research Center, Hawthorne, NY (2001)

Papantonopoulos, G., Modis, K.: A BME solution of the stochastic three-dimensional Laplace equation representing a geothermal field subject to site-specific information. J. Stochastic Environ. Res. Risk Assess. **20**, 23–32 (2006)

Pardo-Igúzquiza, E., Chica-Olmo, M., Atkinson, P.M.: Downscaling cokriging for image sharpening. Remote Sens. Env. **102**, 86–98 (2006)

Paredaens, J., Van den Bussche, J., Van Gucht, D.: First-order queries on finite structures over the reals. SIAM J. Comp. **27**(6), 1747–1763 (1998)

Paredaens, J., Van den Bussche, J., Van Gucht, D.: Towards a theory of spatial database queries. In: Proceedings of the 13th ACM Symposium on Principles of Database Systems, pp. 279–288 (1994)

Parent, C., Spaccapietra, S., Zimányi, E.: Conceptual Modeling for Traditional and Spatio-Temporal Applications: The MADS Approach, p. 466. Springer, Berlin/Heidelberg (2006)

Parent, C., Spaccapietra, S., Zimányi, E.: The MurMur Project: modeling and querying multi-represented spatio-temporal databases. Inf. Syst. **31**, 733–769 (2006)

Park, D., Cho, H., Kim, Y.: A TIN compression method using Delaunay triangulation. Int. J. Geographical Information Science **15**, 255–269 (2001)

Park, R.E., Burgess, E.W., McKenzie, R.D.: The City: Suggestions for Investigation of Human Behavior in the Urban Environment. University of Chicago Press, Chicago, Il (1925)

Partington, E., Wells, D., Manguerra, H.: What's new with EPA/Office of Water's BASINS Water Quality Modeling System. ESRI User's Conference. San Diego, CA http://gis.esri.com/library/userconf/proc01/professional/papers/pap909/p909.htm (2001)

Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Discovering frequent closed itemsets for association rules. In: Proceedings of the International Conference of Database Theory – ICDT '99. Lecture Notes in Computer Science vol. 1540, pp. 398–416 (1999)

Patalaviciute, V., et al.: Using SVG-based Maps for Mobile Guide Systems—A Case Study for the Design, Adaptation, and Usage of SVG-based Maps for Mobile Nature Guide Applications. (2005) Available via http://www.svgopen.org/2005/papers/MapsForMobileNatureGuideApplications/index.html

Patel, J., Chen, Y., Chakka, V.: STRIPES: An Efficient Index for Predicted Trajectories. In: Proceedings of the 2004 ACM SIGMOD, Paris, France, 13–18 June 2004, pp. 637–646

Paterson, C.: The effectiveness of using machine translators to translate German and Portuguese into English when searching for images with English captions. Masters Dissertation, Department of Information Studies, University of Sheffield (2002)

Pathirana, P., Bulusu, N., Jha, S., Savkin, A.: Node localization using mobile robots in delay-tolerant sensor networks. IEEE Trans. Mob. Comp. **4**(4), 285–296 (2005)

Patil, G.P., Bhat, K.S., Kase, M.: Multicriterion prioritization with differential weights, stepwise aggregation, Hasse Diagrams, and Poset cumulative rank frequency operators. In: The 7th Annual International Conference on Digital Government Research, San Diego, CA, May 21–24, 2006

Patil, G.P., Duczmal, L., Haran, M., Patankar, P.: On PULSE: the progressive upper level set scan statistic system for geospatial and spatiotemporal hotspot detection. In: The 7th Annual International Conference on Digital Government Research, San Diego, CA, May 21–24, 2006

Patil, G.P., Taillie, C.: Multiple indicators, partially ordered sets, and linear extensions: multi-criterion ranking and prioritization. Environ. Ecol. Stat. **11**, 199–228 (2004)

Patil, G.P., Taillie, C.: Upper level set scan statistic for detecting arbitrarily shaped hotspots. Environ. Ecol. Stat. **11**, 183–197 (2004)

Pattee, H.H. (ed.): Hierarchy Theory – The Challenge of Complex Systems. Braziller, New York (1973) 156 pp

Patterson, C., Muntz, R., Pancake, C.: Challenges in location-aware computing. IEEE Pervasive Computing **2**(2) 80–89 (2003)

Patwari, N., Ash, J., Kyperountas, S., Moses, R.M., Hero III, A.O., Correal, N.S.: Locating the nodes: Cooperative localization in wireless sensor networks. IEEE Signal Process. Mag. **22**(4), 54–69 (2005)

Patwari, N., Hero III, A.O.: Indirect radio interferometric localization via pairwise distances. In: Third IEEE Workshop on Embedded Networked Sensors (EmNets'06), May (2006), pp. 26–30

Patwari, N., Hero III, A.O.: Manifold learning algorithms for localization in wireless sensor networks. In: IEEE Intl. Conf. on Acoustic, Speech, & Signal Processing (ICASSP'04). **3**, 857–860 (2004)

Patwari, N., Hero III, A.O., Perkins, M., Correal, N., O'Dea, R.J.: Relative location estimation in wireless sensor networks. IEEE Trans. Signal Process. **51**(8), 2137–2148 (2003)

Patwari, N., Hero III, A.O.: Signal strength localization bounds in ad hoc & sensor networks when transmit powers are random. In: Fourth IEEE Workshop on Sensor Array and Multichannel Processing (SAM'06), July (2006)

Paulsen, D. J., Robinson, M.B.: Spatial Aspects of Crime: Theory and Practice. Allyn and Bacon, Boston, MA (2004)

Pauly, A., Schneider, M.: Topological Predicates between Vague Spatial Objects. 9th Int. Symp. on Spatial and Temporal Databases. Angra dos Reis, Brazil (2005)

Pauly, A. Schneider, M.: Topological Reasoning for Identifying a Complete Set of Topological Predicates between Vague Spatial Objects. 19th Florida Artificial Intelligence Research Society (FLAIRS) Conference. (2006) pp. 731–736. Melbourne Beach, Florida, USA

Pauly, A., Schneider, M.: Vague Spatial Data Types, Set Operations, and Predicates. 8th East-European Conf. on Advances in Databases and Information Systems. Budapest, Hungary 22–25 Sep 2004

Pavlidis, T., Horovitz, S.L.: Segmentation of plane curves. IEEE Trans. Comput. **23**(8), 860–870 (1974)

Pavy, N., Rombauts, S., Dehais, P., Mathe, C., Ramana, D.V.V., Leroy, P., Rouze, P.: Evaluation of gene prediction software using a genomic data set: application to Arabidopsis thaliana sequence. Bioinf. **15**(11), 887–899 (1999)

Pawlak, Z., Grzymala-Busse, J., Slowinski, R., Ziarko, R.A.: Rough sets. Communications of the ACM **38**(11), 89–95 (1995)

Pawlak, Z.: Rough sets. Internat. J. Comput. Inform. **11**, 341–356 (1982)

Pawlak, Z.: Rough sets: theoretical aspects of reasoning about data. Theory and decision library. Series D, System theory, knowledge engineering, and problem solving, vol. 9. Kluwer Academic Publishers, Dordrecht, Boston (1991)

PBS&J: Floyd Behavioral Reports (Evacuation Traffic Information System), vol. 2005. PBS&J, Tallahassee, Florida (2005)

PBS&J: Reverse Lane Standards and Its Strategies. PBS&J Southeast United States Hurricane Study, Tallahassee (2000)

PBS&J: Southeast United Sates Hurricane Evacuation Traffic Study: Evacuation Travel Demand Forecasting System, Technical Memorandum 2, Final Report. PBS&J, Tallahassee (2000)

A. P. D. Mourelatos. Events, processes, and states. In: Tedeschi, P., Zaenen, A. (eds.) Tense and Aspect, pp. 191–212. Academic Press, New York (1981)

Peano, G.: Sur une courbe qui remplit toute une air plaine. Mathematishe Annalen **36**, 157–160 (1890)

Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco (1988)

Peddemors, A.J.H., Lankhorst, M.M., de Heer, J.: Presence, Location, and Instant Messaging in a Context-Aware Application Framework. Proc. of the Intl. Conf. on Mobile Data Management (MDM) **2574**, 325–330

Pedersen, T.B., Tryfona, N.: Pre-aggregation in spatial data warehousess. In: Proc. 2001 Int. Symp. Spatial and Temporal Databases (SSTD'01), pp. 460–480. Redondo Beach, CA, July (2001)

Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Wang, J., Chen, Q., Dayal, U., Hsu, M.-C.: Mining sequential patterns by pattern-growth: The prefixspan approach. Proc. of SIGKDD (2004)

Pelanis, M., Saltenis, S., Jensen, C.S.: Indexing the past, present and anticipated future positions of moving objects. ACM Trans. Database Syst. **31**(1), 255–298 (2006). doi: 10.1145/1132863.1132870

Pelechano, N., Badler, N.: Modeling Crowd and Trained Leader Behavior during Building Evacuation. IEEE Computer Graphics and Applications **26**(6), 80–86 (2006)

Peng, M., Xie, L., Pietrafesa, J.: A numerical study of storm surge and inundation in the Croatan-Albemarle-Pamlico Estuary System. Estuarine, Coastal and Shelf Science **59**, 121–137 (2004)

Peng, W., Sijmons, K., Brown, A.: Voronoi Diagram and Delaunay Triangulation Supporting Automated Generalization. Paper presented at 17th ICA/ACI, Barcelona, Spain, 1995

Peng, Z.R., Tsou, M.H.: Internet GIS – Distributed Geographic Information Services for the Internet and Wireless Networks. Wiley Publishing, Hoboken, New Jersey (2003)

Percivall G (2003) OpenGIS Reference Model. OGC project document 03-040. http://portal.opengeospatial.org/files/?artifact_id=3836

Perich, F.: On P2P Data Management in Pervasive Computing Environments, Ph. D. thesis, UMBC, May 2004

Perng, S., Wang, H., Zhang, S., Parker, D.S.: Landmarks: a new model for similarity-based pattern querying in time series databases. In Proc. of IEEE ICDE, pp. 33–42 (2000)

Perreault L., Lapierre C., Plante M., Bachy D., Caron C.: MicroGeomatics: a new paradigm for mobile workforce management. GeoWorld **March**, 26–28 (2006)

Perrizo, W., Jockheck, W., Perera, A., Ren, D., Wu, W., Zhang, Y.: Multimedia data mining using P-trees. In: International Workshop on Multimedia Data Mining (MDM/KDD), Edmonton, Alberta Canada, 2002, pp. 19–29

Perry, A.H., Symons, L.J. (ed.): Highway Meteorology. E & FN Spon (1991)

Pervasive Technology Labs at Indiana University (2004) http://grids.ucs.indiana.edu/ptliupages/. Accessed 20 Aug 2007

Peterson, D.L., Parker, V.T.: Ecological Scale: Theory and Applications. Columbia University Press, New York (1998)

Peterson, M. (ed.) Maps and the Internet, Elsevier, Oxford, 2003, ISBN 0-08-044201-3, 451 pages (2003)

Peterson, M.P.: Interactive and Animated Cartography. Prentice Hall, Englewood Cliffs, New Jersey (1995)

Pettit, C.J.: Use of a collaborative GIS-based planning-support system to assist in formulating a sustainable-development scenario for Hervey Bay, Australia. Environment and Planning B: Planning and Design **32**:523–545 (2005)

Pettit, C., Shyy, T., Stinson, R.: An on-line planning support system to evaluate urban and regional planning scenarios. In: Geertman S., Stillwell, J. (eds.) Planning Support Systems in Practice, pp. 331–347. Springer, New York, NY (2003)

Peucker, T., Chrisman, N.: Cartographic data structures. The American Cartographer **2**(1), 55–69 (1975)

Peuquet, D.J., Duan, N.: An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. Int. J. Geogr. Inf. Syst. **9**, 7–24 (1995)

Pfoser D., Jensen, C.: Capturing the uncertainty of moving-objects representations. In: Proc. SSDBM, 1999

Pfoser, D., Jensen, C.S., Theodoridis, Y.: Novel approaches in query processing for moving object trajectories. In: VLDB, pp. 395–406 (2000)

Pfoser, D., Tryfona, N., Voisard, A.: Dynamic travel time maps – enabling efficient navigation. In: Proc. 18th SSDBM Conference, pp. 369–378 (2006)

Pfoser D., Tyfona, N., Jensen, C.: Indeterminacy and Spatiotemporal Data: basic definitions and case study. Geoinformatica. **9**(3) (2005)

Phoha, S., Eberbach, E., Brooks, R.: Coordination of multiple heterogeneous autonomous undersea vehicles (AUVs) Special Heterogeneous Multi-Robot Systems Issue of Autonomous Robots, October (1999)

Phoha, S., Peluso, E., Culver, R.L.: A high fidelity ocean sampling mobile network (SAMON) simulator tested for evaluating intelligent control of unmanned underwater vehicles. IEEE J. Ocean Eng. Special Edn, June (2001)

Phoha, S., Peluso, E., Stadter, P., Stover, J., Gibson, R.: A mobile distributed network of autonomous undersea vehicles. In: Proceedings of the 24th Annual Symposium and Exhibition of the Association of Unmanned Vehicle Systems International, Baltimore, MD, 3–6 June 1997

Phong, B.T.: Illumination for Computer Generated Images, Comm, vol 18(6):311–317. ACM, New York (1975)

Pick, J. B., Viswanathan, N., Hettrick, W. J.: A dual census geographical information system in the context of data warehousing. In: Proceedings of the Americas Conference on Information Systems. pp. 265–278. Association for Information Systems, Atlanta, Georgia (2000)

Pietrafesa, L.J., Xie, L., Morrison, J., Janowitz, G.S., Pelissier, J., Keeter, K., Neuherz, R.A.: Numerical modelling and computer visualization of the storm surge in and around the Croatan-Albemarle-Pamlico Estuary system produced by Hurricane Emily of August (1993). Mausam **48**, 567–578 (1997)

Pitas, I.: Digital Image Processing Algorithms. Prentice Hall, New York, NY (1993)

Pitoura E., Samaras, G.: Locating objects in mobile computing. IEEE Transactions on Knowledge and Data Engineering. **13**(4) (2001)

Platzman, L., Bartholdi, J.: Spacefilling curves and the planar travelling salesman problem. J. Assoc. Comput. Mach. **46**, 719–737 (1989)

Plewe, B.: GIS ONLINE; Information retrieval, Mapping and the Internet. OnWord Press, Santa Fe, NM (1997)

Podrenek, M.: Aufbau des DLM50 aus dem BasisDLM und Ableitung der DTK50 – Lösungsansatz in Niedersachsen. Kartographische Schriften, Band 6. pp. 126–130. Kirschbaum Verlag, Bonn (2002)

A Policy on Geometric Design of Highways and Streets, 5th edn., American Association of State Highway and Transportation Officials, Washington, D.C. (2001)

Polkowski, L., Skowron, A.: Rough mereology in information systems. a case study: qualitative spatial reasoning. In: Rough set methods and applications: new developments in knowledge discovery in information systems, pp. 89–135. Physica-Verlag GmbH, Heidelberg, Germany, (2000)

Polkowski, L., Skowron, A.: Rough mereology: A new paradigm for approximate reasoning. J. Approx. Reason. **15**(4), 333–365 (1996)

Polkowski, L.: A survey of recent results on spatial reasoning via rough inclusions. In: Bolc, L., Michalewicz, Z., Nishida, T. (eds.) Intelligent Media Technology for Communicative Intelligence, Second International Workshop, IMTCI 2004. Lecture Notes in Computer Science. Springer, Berlin (2004)

Popivanov, I., Miller, R.J.: Similarity search over time series data using wavelets. In: Proceedings of the 18th International Conference on Data Engineering, San Jose, 26 February–1 March 2002, pp. 212–221

Porkaew, K., Lazaridis, I., Mehrotra, S.: Querying mobile objects in spatio-temporal databases. In: SSTD, pp. 59–78 (2001)

Porkaew, K., Lazaridis, I., Mehrotra, S., Winkler, R.: Database Support for Situational Awareness. In: Vassiliop, M.S., Huang, T.S. (eds.) Computer-Science Handbook for Displays – Summary of Findings from the Army Research Lab's Advanced Displays & Interactive Displays Federated Laboratory, Rockwell Scientific Company (2001)

Port Authority of NY and NJ. http://www.panynj.gov. Accessed May 2005

Portugali, J.: Self-Organization and the City. Springer, Berlin (2000)

PostgreSQL Global Development Group (2007) PostgrSQL. http://www.postgresql.org/. Accessed 1 Jan 2007

Pottinger, R.A., Bernstein, P.A.: Merging models based on given correspondences. In: Proceedings of the Very Large Data Bases Conference (VLDB) 2000, Sept. 10–14, Cairo, Egypt

PPDG (2006) Particle Physics Data Grid. http://www.ppdg.net/. Accessed 26 Sept 2006

Prabhakar, S., Xia, Y., Kalashnikov, D., Aref, W., Hambrusch, S.: Query indexing and velocity constrained indexing: Scalable techniques for continuous queries on moving objects. IEEE Trans. Comput. **51**(10), 1124–1140 (2002)

Prasithsangaree, P., Krishnamurthy P., Chrysanthis, P.: On indoor position location with wireless LANs. Paper presented at: The 13th IEEE international symposium on personal, indoor and mobile radio communications (PIMRC 2002), Lisbon, 2002.

Praun, E., Hoppe, H., Webb, M., Finkelstein, A.: Real-time hatching. In: Computer Graphics (Proceedings of ACM SIGGRAPH 2001), pp. 581–586 (2001)

Pred, A.: The choreography of existence: Comments on Hagerstrand's time-geography and its usefulness. Econ. Geogr. **53**, 207–221 (1977)

Prendergrasr, S.: Intergraph's Jim Meadlock Reflects on 30 Years of Graphics Technology. GeoWorld (October 1999), 54–56

Preparata, F.P., Shamos, M.I.: Computational Geometry: An Introduction. Springer-Verlag, Berlin, Heidelberg, New York (1985)

Price, R., Ramamohanarao, K., Srinivasan, B.: Spatio-temporal Extensions to Unified Modeling Language. In: Proceedings of the Workshop on Spatio-temporal Data Models and Languages, IEEE DEXA'99 Workshop, Florence, Italy, 1–3 Sept 1999

Price, R., Tryfona, N., Jensen, C.S.: Extended Spatiotemporal UML: Motivations, Requirements, and Constructs. J. Database Manage. **11** (4), 14–17 (2000)

Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The cricket location-support system. In: 6th ACM MOBICOM, Boston, MA, August (2000)

Proceedings ISPRS Workshop "Laser scanning 2005", Int. Arch. Photogramm. Remote Sensing Spatial Inf. Sci. **36**, http://www.commission3.isprs.org/laserscanning2005/, Vosselman, G., Brenner, C. (Eds.), accessed July 7, 2007

Procopiuc, C.M., Agarwal, P.K., Har-Peled, S.: STAR-tree: An efficient self-adjusting index for moving objects, ALENEX '02: Revised Papers from the 4th International Workshop on Algorithm Engineering and Experiments, pp. 178–193. Springer-Verlag, London, UK (2002)

PROJ.4 (2000) Cartographic Projections Library. http://www.remotesensing.org/proj. Accessed 1 Jan 2007

Purves, R.S., Clough, P., Jones, C.B., Arampatzis, A., Bucher, B., Finch, D., Fu, G., Joho, H., Syed, A.K. Vaid, S., Yang, B.: The design and implementation of SPIRIT: a spatially aware search engine for information retrieval on the Internet. Int. J. Geogr. Inf. Sci. (2007)

Purves, R.S., Jones, C.B.: Geographic Information Retrieval (GIR) editorial. In: special issue Comput. Environ. Urban Syst. **30**(4): 375–377 (2006)

Pushker, R., D'Auria, G., Alba-Casado, J.C., Rodríguez-Valera, F.: Micro-Mar: a database for dynamic representation of marine microbial biodiversity. BMC Bioinf. **6**, 222 (2005)

QGIS (2006) Homepage of the QGIS project. http://qgis.org. Accessed 21 Dec 2006

QGIS Community (2005): QGIS Community – Interview with Frank Wamerdam (2007) http://qgis.org/content/view/58/44/. Accessed 17 Feb 2007

QGIS Community (2007): QGIS Community – Home (2007) http://qgis.org. Accessed 22 Feb 2007

QGIS WMS (2006) QGIS WMS server homepage. http://karlinapp.ethz.ch/qgis_wms/qgis_wms.html. Accessed 21 Dec 2006

Quadstone Paramics: http://paramics-online.com/index.php. Accessed 10 Sept 2007

Quattrochi, D.A., Goodchild, M.F.: Scale in Remote Sensing and GIS. Lewis, Boca Raton, Boston, London, New York, Washington, D.C. (1997)

Quilfen, Y., Chapron, B., Collard, F., M.L Serre.: Calibration/validation of an altimeter wave period model and application to TOPEX/Poseidon and Jason-1 Altimeters. Mar. Geodesy **3/4**, 27, 535–550 (2004)

Quinn, J.: The daytime population of the central business district of Chicago. Review by Breese, G.W. Am. Sociol. Rev.**15**, 827–828 (1950)

Qu, Y., Wang, C., Wang, X.S.: Supporting fast search in time series for movement patterns in multiple scales. In Proc. of ACM CIKM, pp. 251–258 (1998)

Rabbat, M., Nowak, R.: Distributed optimization in sensor networks. In: IPSN '04: Proc. 3rd Int. Symp. on Information processing in sensor networks, pp. 20–27 (2004)

Rafiei, D., Mendelzon, A.O.: Efficient retrieval of similar time sequences using DFT. In: Proceedings of the 5th International Conference on Foundations of Data Organization and Algorithms, Kobe, Japan, 12–13 November 1998

Rafiei, D., Mendelzon, A.: Querying Time Series Data Based on Similarity. IEEE Trans. on Knowl. Data Eng. **12**(5), 675–693 (2000)

Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB J. **10**, 334–350 (2001)

Ramadurai, V., Sichitiu, M.L.: Localization in wireless sensor networks: a probabilistic approach. Proc. of the 2003 International Conference on Wireless Networks (ICWN2003), vol.1, pp. 275–281, Las Vegas, NV, Jan. (2003)

Ramakrishnan, N., Bailey-Kellogg, C., Tadepalli, S., Pandey, V.N.: Gaussian Process for Active Data Mining of Spatial Aggregates. In: Proceedings of the SIAM International Conference on Data Mining (2005)

Ramakrishnan, R.: Database Management Systems. McGraw-Hill, New York (1998)

Ramakrishnan, R., Gehrke, J.: Database Management Systems, 2nd edn. McGraw-Hill, New York, NY, USA (2001)

Ramsey, P.: The State of Open Source GIS. http://www.refractions.net/white_papers/oss_briefing/2006-06-OSS-Briefing.pdf. Accessed 24 Aug 2006

Randall, T.A., Churchill, C.J., Baetz, B.W.: Geographic information system (GIS) based decision support for neighborhood traffic calming. Can. J. Civ. Eng. **32**(1), 86–98 (2005)

Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. In: Nebel, B., Rich, C., Swartout, W. (eds.) Principles of Knowledge Representation and Reasoning. Proceedings of the Third International Conference (KR92), pp. 165–176. Morgan Kaufmann (1992)

Randolph, J.: Environmental Land Use Planning and Management. Island Press, Washington, DC (2004)

Rao, C.R., Suryawanshi, S.: Statistical analysis of shape of objects based on landmark data. Proc Natl Acad Sci U S A. **93**(22), 12132–12136 (1996)

Rao, K., Kumar, D.S.: Spatial decision support system for watershed management. Water Res. Manag. **18**(5), 407–423 (2004)

Rao K.R., Yip, P.: Discrete Cosine Transform: Algorithms, Advantages, Applications. Academic Press Professional, Inc., San Diego (1990)

Raper, J.: Multidimensional Geographic Information Science. Taylor and Francis, London and New York (2000)

Rappaport, T.S.: Wireless Communications: Principles and Practice. Prentice-Hall Inc., New Jersey (1996)

Raskin, R.G., Pan, M.J.: Knowledge representation in the Semantic Web for Earth and Environmental Terminology (SWEET). Comput. Geosci. **31**, 1119–1125 (2005)

Raskin, R.: Guide to SWEET Ontologies. http://sweet.jpl.nasa.gov/guide.doc. Accessed 21 March 2007

Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambrige, MA (2006)

Rasmussen, J., Pejtersen, A.: Virtual Ecology of Work. In: Flack, J., et al. (eds.) Global Perspectives on the Ecology of Human-Machine Systems, p. 121–156. Lawrence Erlbaum Associates, Hillsdale, New Jersey (1995)

Raster GIS http://www.gsd.harvard.edu/pbcote/courses/gsd6322/raster/index.htm Cambridge, MA, USA

Ratcliffe, J.H.: Location Quotients and Force-field Analysis. 7th Annual International Crime Mapping Research Conference (2004)

Ratcliffe, J.H., McCullagh, M.J.: The Perception of Crime Hotspots: A spatial study in Nottingham, UK. In: Crime Mapping Case Studies: Successes in the Field. National Institute of Justice, Washington, DC (1998)

Rathi, A.K., Solanki, R.S.: Simulation of traffic flow during emergency evacuation. Presented at Simulation Conference, 1993

Raubal, M.: Human wayfinding in unfamiliar buildings: a simulation with a cognizing agent. Cognitive Processing **2**(2–3), p. 363–388 (2001)

Raubal, M., Miller, H.J., Bridwell, S.: User centered time geography for location-based services. Geogr. Annaler B **86**, 245–265 (2004)

Raubal, M.: Ontology and epistemology for agent-based wayfinding simulation. Int. J. Geogr. Inf. Sci. **15**(7), 653–665 (2001)

Ravikanth, K.V., Godfrind, A., Beinat, E.: Pro Oracle Spatial: From Professional to Expert. Apress, Berkley (2005). ISBN 1590593839

Ray, J.C.: Crime Prevention through Environmental Design. Sage Publications, Beverly Hills, CA (1977)

R Development Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2004) 3-900051-07-0

Reed, C. (2004) Data integration and interoperability: ISO/OGC standards for geo-information. www.directionsmag.com/article.php?article_id=687. Accessed

Refractions Research (2007) PostGIS. http://postgis.refractions.net/. Accessed 1 Jan 2007

REGIS. Grasslinks 3.1. http://www.regis.berkeley.edu/grasslinks/

Reichenbacher, T.: Adaptive egocentric maps for mobile users. In: Meng, L., Zipf, A., Reichenbacher, T. (eds.) Map-Based Mobile Services. Theories, Methods and Implementations, pp. 141–158. Springer, Heidelberg (2005)

Reichenbacher, T.:, Mobile Cartography Adaptive Visualisation of Geographic Information on Mobile Devices. Verlag Dr. Hut, Munich (2004)

Reid, D.B.: An algorithm for tracking multiple targets. IEEE Trans. Automat. Control **24**, 843–854 (1979)

Reid, H.: Applying Oracle Spatial to a very large insurance problem. Location Intelligence, June 20. Available at www.locationintelligence.net (2006)

Reid, H.: Telematics Detroit, Directions Magazine, May 22 (2005)

Reiners, W.A.: Transport of energy, information and material through the biosphere. Annual Review of Environment and Resources. **28**(1) (2003)

Reiss, A.J., Tonry, M. (eds.): Communities and Crime, Vol 8. University of Chicago Press, Chicago, Il (1986)

Rengert, G.F.: Behavioral Geography and Criminal Behavior. In: Evans, D.J., Herbert, D.T. (eds.) The Geography of Crime. Routledge, London (1989)

Rengert, G.F., Simon, H.: Crime Spillover. Ca. Sage Publications, Beverley Hills (1981)

Resource Description Framework http://www.w3.org/RDF/

Revesz, P., Cai, M.: Efficient querying of periodic spatiotemporal objects. Ann. Math. Artif. Intell. **36**(4), 437–457 (2002)

Revesz, P., Chen, R., Kanjamala, P., Li, Y., Liu, Y., Wang, Y.: The MLPQ/GIS Constraint Database System. In: Proc. ACM SIGMOD International Conference on Management of Data. ACM Press, New York (2000)

Revesz, P., Chen, Y.: Efficient aggregation over moving objects. 10th International Symposium on Temporal Representation and Reasoning, pp. 118–127. IEEE Press, Washington (2003)

Revesz, P.: A closed-form evaluation for Datalog queries with integer (gap)-order constraints. Theor. Comput. Sci. **116**(1), 117–49 (1993)

Revesz, P.: Efficient rectangle indexing algorithms based on point dominance. 12th International Symposium on Temporal Representation and Reasoning, pp. 210–212. IEEE Press, Washington (2005)

Revesz, P., Li, L.: Constraint-based visualization of spatial interpolation data. In: Proc. of the Sixth International Conference on Information Visualization, pp. 563–569. IEEE Press, London, England (2002)

Revesz, P., Li, L.: Constraint-based visualization of spatiotemporal databases. In: Advances in Geometric Modeling, chapter 20, pp. 263–276. John Wiley, England (2003)

Revesz, P., Li, L.: Representation and querying of interpolation data in constraint databases. In: Proc. of the Second National Conference on Digital Government Research, pp. 225–228, Los Angeles, California (2002)

Revesz, P., Li, Y.: MLPQ: A linear constraint database system with aggregate operators. In: Proc. 1st International Database Engineering and Applications Symposium, pp. 132–137. IEEE Press, Washington, DC (1997)

Revesz, P., Wu, S.: Spatiotemporal Reasoning About Epidemiological Data. Artificial Intelligence in Medicine **38**(2), 157–170 (2006)

Revesz, P., Wu, S.: Visualization of recursively defined concepts. In: Proceedings of the 8th International Conference on Information Visualization, pp. 613–621. IEEE Press, Washington, DC (2004)

Revesz, R.Z.: Introduction to Constraint Databases. Springer-Verlag, New York (2002)

Rezende, F.A.V.S., Almeida, R.M.V., Nobre, F.F.: Defining catchment areas for public hospitals in the municipality of rio de janeiro through weighted voronoi diagrams. In Cadernos de Saude Publica, vol. 16, pp. 467–475. (2000)

The RFID Gazette: Wal-Mart Doubling RFID-Enabled Stores. http://www.rfidgazette.org/walmart/, The RFID Journal: Retreived 11/05/2006 (2006)

The RFID Journal: The History of RFID Technology. http://www.rfidjournal.com/article/articleview/1338/1/129/, The RFID Journal: Retreived 10/10/2006 (2006)

The RFID Journal: RFID System Components and Costs. http://www.rfidjournal.com/article/articleview/1336/1/129/, The RFID Journal: Retreived 10/12/2006 (2006)

Riccio, A., Barone, G., Chianese, E., Giunta, G.: A hierarchical Bayesian approach to the spatio-temporal modeling of air quality data. Atmosph. Environ. **40**, 554–566 (2006)

Richards, J.A., Xiuping, J.: Remote Sensing Digital Image Analysis, 4th edn. p. 439. Springer-Verlag, Heidelberg (2005)

Riede, K.: Conservation and modern information technologies: The Global Register of Migratory Species (GROMS). J Int Wildl Law Policy **3**(2), 152–165 (2000)

Rigaux, Ph., Scholl, M., Voisard, A.: Introduction to Spatial Databases: Applications to GIS. Morgan Kaufmann, (2000)

Rigaux, P., Scholl, M., Segoufin, L., Grumbach, S.: Building a Constraint-Based Spatial Database System: Model, Languages, and Implementation. Inf. Syst. **28**(6), 563–595 (2003)

Rigaux, P., Scholl, M., Voisard, A.: Spatial Databases: With Application to GIS. 2nd edn. Morgan Kaufmann Publishers, San Francisco, CA, United States, May 2001

Rios-Insua, D.: Sensitivity Analysis In Multi-Objective Decision Making. Springer-Verlag, Berlin (1990)

Ripley, B.: Spatial Statistics. Wiley and Sons, Inc., New York (1981)

Riscure: Privacy issues with new digital passport. http://www.riscure.com/2_news/passport.html, Riscure BV: Retreived 11/08/2006 (2006)

Ristic, B., Arulampalm, S., Gordon, N.: Beyond the Kalman Filter: Particle Filters for Tracking Applications. Artech House, Boston (2004)

Rittman, M.: GIS-enabling your Oracle data warehouse. DBAzine.com. April 18. Available at www.dbazine.com (2006)

Rittman, M.: An introduction to Oracle warehouse builder 10g. DBAzine.com. August 19. Available at www.dbazine.com (2006)

Rizzoli, A., Merler, S., Furlanello, C., Genchi, C.: Geographical information system and bootstrap aggregation (bagging) of tree-based classifiers for lyme disease risk prediction in trentino, italian alps. J. Med. Entomol. **39**(3), 485–492 (2002)

Road Traffic Technology: http://www.roadtraffictechnology.com/contractors/traffic_man/ptv_ag/ptv_ag2.html. Accessed 10 Sept 2007

Robert, C.: The Bayesian Choice, 2nd edn. Springer, New York (2001)

Roberti, M.: DOD Releases Final RFID Policy. http://www.rfidjournal.com/article/articleview/1080/1/1/, The RFID Journal: Retreived 10/11/2006 (2004)

Robinson, G.K.: That BLUP is a Good Thing: The Estimation of Random Effects, Statistical Science, vol. 6, pp. 15–32 (1991)

Robinson, A.H., Morrison, J.L., Muehrcke, P.C., Kimerling, A.J., Guptill, S.C.: Elements of Cartography. Wiley, New York (1995)

Robinson, J.T.: The K-D-B-tree: a search structure for large multidimensional dynamic indexes. In: Proc. of the 1981 ACM SIGMOD international conference on Management of data (SIGMOD'81), Ann Arbor, Michigan (1981)

Rocha, L.V., Edelweiss, N., Iochpe, C.: GeoFrame-T: a temporal conceptual framework for data modeling. In: ACM Symposium on Advances in Geographic Information Systems, Atlanta, 9–10 Nov 2001, pp. 124–129

Roddick, J.F., Hoel, E., Egenhofer, M.E., Papadias, D., Salzberg, B.: Spatial, temporal and spatio-temporal databases – hot issues and directions for phd research. SIGMOD Rec., **33**(2), 126–131 (2004)

Roddick, J.F., Spiliopoulou M.: A bibliography of temporal, spatial and spatio-temporal data mining research. SIGKDD Explor. Newsl. **1**, 34–38 (1999)

Rodriguez, A., Egenhofer, M.: Assessing Similarity Among Geospatial Feature Class Definitions. In: Vckorski, A., Brasel, K.E., Schek, H.J. (eds.) Lecture Notes in Computer Science 1580, pp. 189–202. Springer, Berlin (1999)

Rodriguez, A., Egenhofer, M.: Determining Semantic Similarity Among Entity Classes from Different Ontologies. IEEE Transactions on Knowledge and Data Engineering **15**, 442–456 (2003)

Rodriguez, M., Egenhofer, M., Rugg, R.: Comparing geospatial entity classes: An asymmetric and context-dependent similarity measure. Int. J. Geogr. Inf. Sci. **18**(3), 229–256 (2004)

Roensdorf, C.: Achieving and Maintaining Interoperability of Spatial Data, FIG working week proceedings, Cairo. http://www.fig.net/pub/cairo/papers/ts_14/ts14_02_ronsdorf.pdf (2004)

Roensdorf, C.: PAI2: Achieving Geometric Interoperability of Spatial Data. In: EuroSDR official workshop publication. no. 49, Dublin: http://www.eurosdr.net/km_pub/no49/html/PAI2/index.htm (2005)

Rogers, E.M.: Diffusion of Innovations, 4th edn. Free Press, New York (1995)

Romano, N., Palladino, M.: Prediction of soil water retention using soil physical data and terrain attributes. J. Hydro. **265**, 56–75 (2002)

Ronald, C.V.: Situational Crime Prevention. Harrow and Heston (1992)

Roos, T., Myllymaki, P., Tirri, H., Misikangas, P., Sievanen, J.: A Probabilistic Approach to WLAN User Location Estimation. Int. J. Wirel. Inf. Netw. **9**(3), 155–164 (2002)

Roos, T., Myllymäki, P., Tirri, H.: A statistical modeling approach to location estimation. IEEE Trans. Mobile Comput. **1**, 59–69 (2002)

Roques, P., Vallée, F.: UML en Action – De l'analyse des besoins à la conception en Java, 2nd edn. Eyrolles, Paris (2002)

Roque, W.L., Choset, H.: The green island formation in forest fire modeling with voronoi diagrams. In: the 3rd Center for Geometric Computing Workshop on Compuational Geometry, (1998)

Rosenfeld, A.: Computer image analysis: an emerging technology in the service of society. TR-1177, MCS-79-23422, Comput. Sci. Tech. Rep. University of Maryland (1982)

Rosenkrantz, D.J., Stearns, R.E., Lewis II, P.M.: An analysis of several heuristics for the traveling salesman problem. SIAM J. Comput. **6**, 563–581 (1977)

Ross, D.T., Rodriguez, J.E.: Theoretical foundations for the computer aided design system, pp. 305–322. Proceedings of AFIPS 23, SJCC, Detroit (1963)

Rossmo, D.K.: Geographic Profiling. CRC Press, Boca Raton, FL (2000)

Rossmo, D.K.: Overview: multivariate spatial profiles as a tool in crime investigation. In: Block, C.R., Dabdoub, M., Fregly, S. (eds.) Crime Analysis Through Computer Mapping, pp. 65–97. Police Executive Research Forum, Washington, DC (1995)

Rosson, M.B.: The end of users. In: Proceedings of the OOPSLA (2005) Conference (keynote presentation), San Diego, California, October 16–20, 2005

Ross, R.: A logical basis of the sanitary policy of mosquito reduction. Science **22**, 689–699 (1905)

Rost, R.J.: OpenGL Shading Language, 2nd edn. Addison Wesley, Boston (2006)

Rottenbacher, C.: Emotional Correspondence Links. CORP 2005 (2005)

Rottenbacher, C.: Motion Increeases Emotional Correspondence in Geocollaboration. GIScience 2004. Baltimore, (2004)

Rottenbacher, C.: Presence in the Planning Process. GEOS 2004. Brasilien, (2004)

Rottenbacher, C.: Shared Knowledge Construction in Heterogeneous Groups. CORP 2006 (2006)

Roussopoulos, N., Kelley, S., Vincent, F.: Nearest neighbor queries. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 71–79 (1995)

Roussopoulos, N., Leifker, D.: Direct spatial search on pictorial databases using packed r-trees. SIGMOD Record **14**(4):17–31 (1985)

Roussos, G., Kourouthanasis, P., Spinellis, D., Gryazin, E., Pryzbliski, M., Kalpogiannis, G., Giaglis, G.: Systems architecture for pervasive retail, In: SAC '03: Proceedings of the 2003 ACM symposium on Applied computing (ACM Press, New York 2003)

Rowshan, S.: A Guide to Highway Vulnerability Assessment for Critical Asset Identification and Protection. FHWA, pp. 47 (2002)

Rowshan, S., Smith, M.C., Krill, S., Seplow, J.E., Sauntry, W.C.: A State DOT Guide to Highway Vulnerability Assessment, AASHTO (2003)

Roy, B.: Multicriteria methodology for decision aiding. Kluwer Academic Publishers, Dordrecht (1996)

Rue, H., Held, L.: Gaussian Markov Random Fields: Theory and Applications. Chapman & Hall/CRC, New York (2005)

Rui, Y., Huang, T., Chang, S.: Image retrieval: current techniques, promising directions and open issues. J. Vis. Commun. Image Represent. **10**(4): 39–62 (1999)

Rui, Y., Huang, T.S.: Image retrieval: past, present and future. Invited paper in: International Symposium on Multimedia Information Processing, Taipei, Taiwan, December 11–13, 1997

Rule Markup Language (2002) http://www.ruleml.org/

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W.: Object-Oriented Modeling and Design. Prentice-Hall, Englewood Cliffs (1991)

Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language, Reference Manual, 2nd edn. Addison Wesley, Boston, MA, USA (2005)

Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language Reference Manual. Addison-Wesley, Reading, MA (1999)

Ruoming J., Agrawal, G.: Efficient decision tree construction on streaming data. In: ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD), pp. 571–576 (2003)

Rusak Masur, E., Castner, H.W.: Horton's ordering scheme and the generalisation of river networks. Cartogr. J. **27**, 104–112 (1992)

Rushton, G.: Public health, GIS, and spatial analytic tools. Ann. Rev. Pub. Health **24**, 43–56 (2003)

Russell, C.A., Smith, D.A., Waller, L.A., Childs, J.F., Real, L.A.: *A priori* prediction of disease invasion dynamics in a novel environment. Proceedings of the Royal Society: Biol. Sci. **271**:1534, 21–25 (2004)

Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. 2nd edn. Prentice Hall Series in Artificial Intelligence. Prentice Hall, London (2003)

Saalfeld, A.: Conflation: Automated Map Compilation. Int. J. Geogr. Inf. Sci. **2**(3), 217–228 (1988)

Saalfeld, A.: Conflation: Automated Map Compilation. US Bureau of Census, SRD Research Report Number: Census/SRD/RR-87/24. http://www.census.gov/srd/papers/pdf/rr87-24.pdf (1997)

Saaty, T.L.: The analytic hierarchy process. McGraw-Hill, New York (1980)

Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and Analysis of Computer Experiments. Stat. Sci. **4**(4), 409–435 (1989)

Safiolea, E., Bedient, P.B., Vieux, B.E.: Assessment of the Relative Hydrologic Effects of Land Use Change and Subsidence Using Distributed Modeling. In: Glenn E. Moglen (ed.) ASCE, Engineering, Ecological, and Economic Challenges Watershed, July 19–22, 2005. Williamsburg, Virginia, 178, 87 (2005)

Sagan, H.: Space Filling Curves. Springer, Berlin (1994)

Saglio, J.-M., Moreira, J.: Oporto: A realistic scenario generator for moving objects. In: Proc. DEXA'99 Workshop on Spatio-temporal Data Models and Languages (STDML), pp. 426–432. Florence, Italy (1999)

Saha, S.: Image compression—from DCT to wavelets: a review. http://www.acm.org/crossroads/xrds6-3/sahaimgcoding.html

Said, A., Pearlman, W.A.: A new, fast, and efficient image codec based on set partitioning in hierarchical trees. IEEE Trans. Circuits and Systems for Video Technology **6**, 243–250 (1996)

Saito T., Takahashi T.: Comprehensible Rendering of 3-D Shapes. Computer Graphics (Proceedings of ACM SIGGRAPH '90), pp. 197–206 (1990)

Salmenkivi, M.: Efficient mining of correlation patterns in spatial point data. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) Knowledge Discovery in Databases: PKDD-06, Berlin, Germany, Proceedings. Lecture Notes in Computer Science 4213, pp. 359–370. Springer, Berlin (2006)

Salmenkivi, M.: Evaluating attraction in spatial point patterns with an application in the field of cultural history. In: Proceedings of the 4th IEEE International Conference on Data Mining (2004)

Saltelli, A., Chen, K., Scott, E.M.: Sensitivity Analysis. Chichester, New York (2000)

Saltelli, A., Tarantola, S., Campolongo, F., Ratto, M.: Sensitivity Analysis in Practice: a Guide to Assessing Scientific Models. Wiley, Hoboken, NJ (2004)

Šaltenis, S., Jensen, C.S.: Indexing of Moving Objects for Location-Based Services. In: Proceedings of ICDE, pp. 463–472. San Jose, California, USA, 26 Feb–1 Mar 2002

Saltenis, S., Jensen, C.S., Leutenegger, S.T., Lopez, M.A.: Indexing the positions of continuously moving objects. In: SIGMOD 00: Proceedings of the 2000 ACM SIGMOD international conference on Management of data, pp. 331–342, Dallas, Texas, 16–18 May 2000

Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM **18**, 613–620 (1975)

Salway, R., Wakefield, J.: Sources of bias in ecological studies of non-rare events. Environ. Ecol. Stat. **12**, 321–347 (2005)

Salzberg, B., Tsotras, V.J.: Comparison of access methods for time-evolving data, ACM Comput. Surv. **31**(2), 158–221 (1999)

Samal, A., Seth, S., Cueto, K.: A feature-based approach to conflation of geospatial sources. Int. J. GIS **18**(5), 459–589 (2004)

Samarati, P.: Protecting respondents' identities in microdata release. IEEE Transactions on Knowledge and Data Engineering **13**(6):1010–1027 (2001)

Samarati, P., Sweeney, L.: Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression (Technical Report SRI-CSL-98-04). Computer Science Laboratory, SRI International (1998)

Samet, H.: Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS. Addison-Wesley, Reading, MA (1990)

Samet, H.: The Design and Analysis of Spatial Data Structures. Addison-Wesley Publishing Company, San Francisco (1990)

Samet, H.: Foundations of Multidimensional and Metric Data Structures. Morgan Kaufmann, San Francisco, CA (2006)

Samet, H.: The quadtree and related hierarchical data structures. ACM Computing Surveys **16**(2):187–260 (1984)

Sampson, R.J., Raudenbush, S.R., Earls, F.: Neighborhoods and Violent Crime: A Multilevel Study of Collective Efficacy. Science **227**, 918–924 (1997)

Samra, J.S., Gill, H.S., Bhatia, V.K.: Spatial Stochastic Modeling of Growth and Forest Resource Evaluation. Forest Science **35**, 663–676 (1989)

Sandau, R., Braunecker, B., Driescher, H., Eckart, A., Hilbert, S., Hutton, J., Kirchhofer, W., Lithopoulos, E., Reulke, R., Wicki, S.: Design principles of the LH Systems ADS40 airborne digital sensor. IAPRS, vol. XXXIII, part B1, Amsterdam, pp. 258–265, digitally on CD-Rom (2000)

Sanderson, E.W., Jaiteh, M., Levy, M.A., Redford, K.H., Wannebo, A.V., Woolmer, G.: The human footprint and the last of the wild. Bioscience **52**:891–904 (2002)

Sanderson, M., Kohler, J.: Analyzing geographic queries. In: The proceedings of Workshop on Geographic Information Retrieval SIGIR (2004)

Sanjay E. Weis, S., Weis, S.A., Engels, D.W.: RFID Systems and Security and Privacy Implications, In: Workshop on Cryptographic Hardware and Embedded Systems, Lecture Notes in Computer Science, vol. 2523, pp. 454–470 (2002) http://citeseer.ist.psu.edu/sarma02rfid.html

Sankaranarayanan, J., Alborzi, H., Samet, H.: Efficient Query Processing on Spatial Networks, In: Proc. ACM International Workshop on Geographical Information Systems, pp. 200–209 (2005)

Santhi, C., Arnold, J.G., Williams, J.R., Hauck, L.M., Dugas, W.A.: Application of a watershed model to evaluate management effects on point and nonpoint source pollution. Transactions of the American Society of Agricultural Engineers **44**(6), 1559–1570 (2001)

Santini, S., Jain, R.: Similarity Measures. IEEE Trans. Pattern Analysis and Machine Intelligence **21**(9), 87–883 (1999)

Santi, P.: Topology control in wireless ad hoc and sensor networks. ACM Comput. Surveys **37**, 164–194 (2005)

Santner, T.J., Williams, B.J., Notz, W.I.: The Design and Analysis of Computer Experiments. Springer, New York (2003)

Santos, P., Bennett, B., Sakellariou, G.: Supervaluation semantics for an inland water feature ontology. In: Kaelbling, L.P., Saffiotti, A. (eds.): Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05), pp. 564–569. Professional Book Center, Denver (2005)

Sarawagi, S., Agrawal, R., Megiddo, N.: Discovery-driven exploration of OLAP data cubes. In: Proc. Int. Conf. of Extending Database Technology (EDBT'98), pp. 168–182. Valencia, Spain, Mar. (1998)

Sarjakoski, L.T., Koivula, T., Sarjakoski, T.: Knowledge-Based Map Adaptation for Mobile Map Services. In: Gartner, G., Cartwright, W., Peterson, M.P. (eds.) Location Based Services and Tele-Cartography. Lecture Notes in Geoinformation and Cartography. Springer, Berlin Heidelberg New York (2007), pp. 247–264

Sarjakoski, L.T., Sarjakoski, T.: A use case based mobile GI service with embedded map generalisation. Paper presented at the ICA Workshop on Generalization and Multiple Representation, Leicester, 20–21 August 2004. Available via: http://ica.ign.fr/Leicester/paper/Sarjakoski-v2-ICAWorkshop.pdf. Accessed 12 Feb 2007

Sarjakoski, T., Nivala A.-M.: Adaptation to context — a way to improve the usability of mobile maps. In: Meng, L., Zipf, A., Reichenbacher, T. (eds.) Map-Based Mobile Services. Theories, Methods and Implementations, pp. 7–123. Springer, Heidelberg (2005)

Sarjakoski, T., Sarjakoski, L.T.: The GiMoDig public final report. Geospatial info-mobility service by real-time data-integration and generalisation, GiMoDig-project, IST-2000-30090, Deliverable D1.2.31, Public EC report, 70 pages. http://gimodig.fgi.fi/deliverables.php (30/03/07), 2005

Sarjakoski, T., Sarjakoski, L.T.: A Real-Time Generalisation and Map Adaptation Approach for Location-Based Services. In: Mackaness, W.A., Ruas, A., Sarjakoski, L.T. (eds.) Generalisation of Geographic Information: Cartographic Modelling and Applications. International Cartographic Association, Elsevier (2007), pp. 137–159

Satish, K.P.: Technological frames of stakeholders shaping the SDI implementation: A case study from India. Inf. Technol. Dev. **12**, 311–331 (2006)

Savabi, M.R., Flanagan, D.C., Hebel, B., Engel, B.A.: Application of WEPP and GIS-grass to a small watershed in indiana. J. Soil Water Conserv. **50**(5), 477–483 (1995)

Savage, J.F.: Anticipating the Future of AM/FM Systems: The Benefits of Distributed Relational Technology. Proceedings of Executive Management Symposium, AM/FM International, Keystone, Colorado, August (1991)

Savage, S., Voelker, G.M., Paxson, G.V.V., Weaver, N.: Center for Internet Epidemiology and Defenses. http://www.ccied.org/

Savarese, C., Rabaey, J.M., Beutel, J.: Locationing in distributed ad-hoc wireless sensor networks. In: ICASSP, May 2001, pp. 2037–2040 (2001)

Savarese, D.F.: A Trio of Quadtrees. JAVAPro 7 November (2003)

Savelieva, E., Demyanov, V., Kanevski, M., Serre, M.L., Christakos, G.: BME-based uncertainty assessment of the Chernobyl fallout. Geoderma **128**, 312–324 (2005)

Savitsky, B.G., Lacher, T.E., Jr. (eds.): GIS Methodologies for Developing Conservation Strategies. Columbia University Press (1998)

Savitsky, B.G.: Overview of Gap Analysis. In: Savitsky, B.G., Lacher Jr., T.E. (eds.) GIS Methodologies for Developing Conservation Strategies., pp. 151–157. Columbia University Press (1998)

Savvides, A., Garber, W., Adlakha, S., Moses, R., Srivastava, M.B.: On the error characteristics of multihop node localization in ad-hoc sensor netwoks. In: 2nd Intl. Workshop on Inform. Proc. in Sensor Networks, April 2003

Savvides, A., Han, C.C., Srivastava, M.B.: Dynamic fine-grained localization in ad-hoc sensor networks. In: Savvides, A., Han, C.C., Srivastava, M.B. (eds.) Proceedings of the Fifth International Conference on Mobile Computing and Networking (Mobicom 2001), Rome, Italy, July 2001, pp. 166–179

Savvides, A., Park, H., Srivastava, M.B.: The bits and flops of the n-hop multilateration primitive for node localization problems. In: Intl. Workshop on Sensor Nets. & Apps, Sept 2002, pp. 112–121 (2002)

Sawhill, J.C.: Creating biodiversity partnerships: the Nature Conservancy's perspective. Environ. Manag. **20**:789–792 (1996)

Saxena, M.: Does GIS and RFID Amalgamation Work? http://www.geoplace.com/uploads/FeatureArticle/0505tt.asp, Geoplace: Retreived 10/10/2006 (2005)

Sayed, A.H., Tarighat, A., Khajehnouri, N.: Network-based wireless location. IEEE Signal Process. Mag. **22**(4), 24–40

Scalzo, B.: Oracle DBA Guide to Data Warehousing and Star Schemas, 1st edn. Prentice Hall, New York (2003)

Scarponcini, P.: Generalized Model for Linear Referencing in Transportation, GeoInformatica **6**(1), pp. 35–55 (2002)

Schabenberger, O., Gotway, C.A.: Statistical Methods for Spatial Data Analysis (Texts in Statistical Science Series). Chapman and Hall/CRC, Boca Raton (2004)

Schachter, I.: Decomposition of polygons into convex sets. IEEE Transactions on Computers **27**(11), 1078–1082 (1978)

Schaefer, R.-P., Thiessenhusen, K.-U., Wagner, P.: A traffic information system by means of real-time floating-car data. In: Proc. ITS World Congress, Chicago (2002)

Schafer, W.A., Ganoe, C.H., Xiao, L., Coch, G., Carroll, J.M.: Designing the next generation of distributed, geocollaborative tools. Cartography Geogr. Inf. Sci. **32**, 81–100 (2005)

Schär, C., Vasilina, L., Pertziger, F., Dirren, S.: Seasonal runoff forecasting using precipitation from meteorological data assimilation systems. J. Hydrometeorol. **5**(5), 959–973 (2004)

Scheiner, S.M., Gurevitch, J.: Design and Analysis of Ecological Experiments, 2nd edn. Oxford University Press, Oxford, UK (2001)

Schelling, T.: Micromotives and Macrobehavior, W.W., Norton (1978)

Scheuermann, P., Shim, J., Vingralek, R.: WATCHMAN: A data warehouse intelligent cache manager. In: Proceedings of 22nd International Conference on Very Large Data Bases (VLDB), Mumbai, India, 3–6 Sept 1996, pp. 51–62

Schèurmann, J.: Pattern classification: a unified view of statistical and neural approaches. Wiley, New York, xvii, 373 p (1996)

Schiewe, J.: Effect of lossy data compression techniques of geometry and information content of satellite imagery. IAPRS. **32**, 540–544 (1998)

Schiller, J., Voisard, A.: Location-Based Services. Morgan Kaufmann Publishers, San Francisco, CA (2004)

Schmidt, A.M., Gelfand, A.E.: A Bayesian coregionalization approach for multivariate pollutant data. J. Geophys. Res. **108**, 8783 (2003)

Schneider, B.: GIS Functionality in Multimedia Atlases: Spatial Analysis for Everyone. In: Proceedings of the 20th International Cartographic Conference Beijing, 6–10 August 2001, pp. 829–840

Schneider, B.: Integration of Analytical GIS-functions in Multimedia Atlas Information Systems. In: Proceedings of the 19th International Cartographic Congress Ottawa, 14–21 August 1999, pp. 243–250

Schneider-Hufschmidt, M., Kühme, T., Malinowski, U.(eds.): Adaptive user interfaces: principles and practice. North-Holland, Amsterdam (1993)

Schneider, M., Behr, T.: Topological Relationships between Complex Spatial Objects. ACM Trans. on Database Sys. (TODS) **31**, 39–81 (2006)

Schneider, M.: Modelling spatial objects with undetermined boundaries using the Realm/ROSE approach. In: Burrough, P.A., Frank, A.U. (eds.) Geographic Objects with Indeterminate Boundaries, GISDATA Series, pp. 141–152. Taylor & Francis, London (1996)

Schneider, M.: Spatial data types for database systems: finite resolution geometry for geographic information systems. Lecture Notes in Computer Science, vol. 1288. Springer Verlag, Berlin (1997)

Schneider, M.: Uncertainty management for spatial data in databases: fuzzy spatial data types. In: Guting, R.H., Papadias, D., Lochovsky, F. (eds.) Advances in Spatial Databases – Sixth International Symposium, SSD '99. Lecture Notes in Computer Science, vol. 1651, pp. 330–351. Springer, Berlin (1999)

Schreck, T., Chen, Z.: Branch grafting method for r-tree implementation. J. Syst. Softw. **53**(1):83–93 (2000)

Schroeder, M.: Fractals, Chaos, Power Laws: Minutes From an Infinite Paradise. W.H. Freeman and Company, NY (1991)

Schut, P., Whiteside, A. (ed.) OpenGIS Web Processing Service. OGC project document 05-007r4. OGC (2005)

Schwartz, P., Randall, D.: An Abrupt Climate Change Scenario and Its Implications for United States National Security. Report to the Pentagon. www.mindfully.org/Air/2003/Pentagon-Climate-Change1oct03.htm. (accessed December 23, 2006) (2003)

Schwarz, K.-P., Chapman, M.E., Cannon, E., Gong, P.: An integrated INS/GPS approach to the georeferencing of remotely sensed data. Photogramm. Eng. Remote Sensing **59**, 1667–1674 (1993)

Schweizer, P.: IPK Gatersleben. www.ipk-gatersleben.de/en/02/04/04/ (2007). Accessed 05 March 2007

Schwiegerling, J., Greinvenkamp, J.E., Miller, J.M.: Representation of videokeratoscopic height data with zernike polynomials. J. Opt. Soc. Am. A **12**(10), 2105–2113 (1995)

SciDAC: Scientific data management center. http://sdm.lbl.gov/sdmcenter/ (2002)

Scott, J.M., Csuti, B., Jacobi, J.D., Estes, J.E.: Species richness: a geographic approach to protecting future biological diversity. Bioscience **37**:782–788 (1987)

SDSS. SkyServer. http://skyserver.sdss.org/dr1/en/

Searby, S.: Personalisation – an overview of its use and potential. BT Tech. J. **21**(1), 13–19 (2003)

Sebastiano, C., Ciurana, M., Barcelo,F.: Indoor tracking in wlan location with toa measurements. In: MobiWac 2006 (2006)

SECONDO System. Available for download at http://www.informatik.fernuni-hagen.de/import/pi4/Secondo.html/

Sederberg, T.W., Greenwood, E.: A physically based approach to 2-D shape blending. Comput. Graphics **26**, 25–34 (1992)

Sedgewick, R.: Algorithms in C++: Part 5 Graph Algorithms, 3rd edn. Addison-Wesley, Boston (2002)

Seedahmed, G., Martucci, L.: Automated Image Registration Using Geometrical Invariant Parameter Space Clustering (GIPSC). In: Proceedings of the Photogrammetric Computer Vision. Graz, Austria (2002)

Seeger, B., Kriegel, H.-P.: The buddy tree: an efficient and robust access method for spatial data bases. In: Proceedings of 16th international conference on Very large databases (VLDB'90), Brisbane, Australia (1990) pp. 590–601

Seidel, A.: Way-Finding in Public Spaces: The Dallas/Fort Worth, USA Airport. 20th International Congress of Applied Psychology. Edinburgh, Scotland (1982)

Sellis, T.: Chorochronos – research on spatiotemporal database systems. In: DEXA '99: Proceedings of the 10th International Workshop on Database & Expert Systems Applications, p. 452. IEEE Computer Society, Washington, DC, USA (1999)

Sellis, T.K.: Research issues in spatio-temporal database systems. In: SSD '99: Proceedings of the 6th International Symposium on Advances in Spatial Databases, pp. 5–11. Springer, London (1999)

Sellis, T.K., Roussopoulos, N., Faloutsos, C.: The R+tree: A dynamic index for multi-dimensional objects. Proceedings of 13th International Conference on Very Large Data Bases (VLDB 1987), Brighton, England, Sept 1987

Semantic Web Rule Markup Language http://www.daml.org/2003/11/swrl/

Senior Citizens From Houston Die When Bus Catches Fire, Washington Post Staff Writer, Saturday, September 24 (2005); Page A09. [Also available online at: http://www.washingtonpost.com/wp-dyn/content/article/2005/09/23/AR2005092300505.html]

Sentz K, Ferson S (2002) Combination of Evidence in Dempster-Shafer belief theory. Available via SANDIA Technical Report SAND2002–0835. http://www.sandia.gov/epistemic/Reports/SAND2002-0835.pdf

Serre, M.L., Christakos, G., Howes, J., Abdel-Rehiem, A.G.: Powering an Egyptian air quality information system with the BME space/time analysis toolbox: Results from the Cairo baseline year study. In: Monestiez, P., Allard, D., Froidevaux, R. (eds.) Geostatistics for Environmental Applications, pp. 91–100. Kluwer, Dordrecht (2001)

Serre, M.L., Christakos, G.: Modern Geostatistics: Computational BME in the light of uncertain physical knowledge–The Equus Beds study. Stochastic Environ. Res. Risk Assess. **13**, 1–26 (1999)

Serre, M.L., Kolovos, A., Christakos, G., Modis, K.: An application of the holistochastic human exposure methodology to naturally occurring arsenic in Bangladesh drinking water. Risk Anal. **23**, 515–528 (2003)

Sester, M., Brenner, C.: Continuous generalization for visualization on small mobile devices. In: Peter Fisher (ed.) Developments in Spatial Data Handling–11th International Symposium on Spatial Data Handling, pp. 469–480. Springer Verlag, Berlin, Heidelberg, New York (2004)

Shachter, R., Peot, M.: Decision making using probabilistic inference methods. In: Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence, pp. 275–283 (1992)

Shadbolt, N., Berners-Lee, T., Hall, W.: The semantic web revisited. IEEE Intell. Syst. **21**, 96–101 (2006)

Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)

Shahabi, C., Tian, X., Zhao, W.: TSA-tree: a wavelet based approach to improve the efficiency of multi-level surprise and trend queries. In: Proceedings of the 12th International Conference on Scientific and Statistical Database Management, Berlin, Germany, 26–28 July 2000, pp. 55–68

Shamsi, U.: GIS Tools for Water, Wastewater, and Stormwater Systems, p. 392, ISBN: 0-7844-0573-5. American Society of Civil Engineers (ASCE Press), Reston Virginia (2002)

Shanahan, M.: Default reasoning about spatial occupancy. Artif. Intell. **74**(1),147–163 (1995)

Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**, 379–423 & 623–656 (1948)

Shapiro, J.M.: Embedded image coding using zerotrees of wavelet coefficients. IEEE Trans. Signal Processing **41**, 3445–3462 (1993)

Sharifzadeh, M., Kolahdouzan, M., Shahabi, C.: The Optimal Sequenced Route Query. VLDB Journal (2007)

Sharifzadeh, M., Shahabi, C.: Additively Weighted Voronoi Diagrams for Optimal Sequenced Route Queries. In: Proceedings of the 3rd International Workshop on Spatio-temporal Database Management (STDBM'06), September (2006). CEUR Workshop Proceedings

Sharifzadeh, M., Shahabi, C.: Processing Optimal Sequenced Route Queries using Voronoi Diagrams. Submitted for review

Sharifzadeh, M., Shahabi, C.: The Spatial Skyline Queries. In: Proceedings of the 32nd International Conference on Very Large Data Bases: VLDB'06. pp. 751–762 September (2006)

Sharifzadeh, M., Shahabi, C.: Supporting spatial aggregation in sensor network databases. In: Proceedings of the 12th annual ACM international workshop on Geographic information systems, Washington DC, USA, 12–13 Nov 2004, pp. 166–17. ACM Press, New York (2004)

Sharma, C.: Wireless Internet Enterprise Applications, Wiley Tech Brief Series. Wiley Computer Publishing, New York, NY (2001)

Shaw, C.R., McKay, H.D.: Juvenile Delinquency and Urban Areas. University of Chicago Press, Chicago, Il (1942)

Shawney, H., Hafner, J.: Efficient Color Histogram Indexing. In: Proc. International Conference on Image Processing, pp. 66–70. IEEE Computer Society, Los Alamitos, CA, USA (1994)

Sheeren, D., Mustière, S., Zucker, J.D.: Consistency assessment between multiple representations of geographical databases: a specification-based approach. In: Proceedings of the 11th International Symposium on Spatial Data Handling, Leicester, UK (2004)

Sheffi, Y., Mahmassani, H.S., Powell, W.B.: A Transportation Network Evacuation Model. Transp. Res., Part A **16**, 209–218 (1982)

Shekhar, S., and Chawla, S.: Spatial Databases – A Tour. Horton, M. (ed). Pearson Education, Inc., Upper Saddle River, NJ, USA (2003)

Shekhar, S., Chawla, S.: Spatial Databases: A Tour. Prentice Hall, Upper Saddle River, NJ (2003)

Shekhar, S., Chawla, S.: A Tour of Spatial Databases. Prentice Hall, Upper Saddle River, New Jersey (2002)

Shekhar, S., Fetterer, A., Goyal, B.: Materialization Trade-Offs in Hierarchical Shortest Path Algorithms. In: SSTD, pp. 94–111. Springer, Berlin (1997)

Shekhar, S., Huang, Y.: Discovering spatial co-location patterns: a summary of results. In: Proceedings of 7th International Symposium on Advances in Spatial and Temporal Databases (SSTD 2001). Springer-Verlag, Redondo Beach, California (2001)

Shekhar, S., Huang, Y., Djugash, J., Zhou, C.: Vector map compression: a clustering approach. In: Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems, McLean, Virginia, 8–9 Nov 2002

Shekhar, S., Kohli, A., Coyle, M.: Path Computation Algorithms for Advanced Traveller Information System (ATIS). In: ICDE, Vienna, Austria, 1993. IEEE Computer Society, pp. 31–39 (1993)

Shekhar, S., Liu, D.R.: CCAM: A Connectivity-Clustered Access Method for Networks and Network Computations. IEEE Trans. Knowl. Data Eng. **9**(1), 102–119 (1997)

Shekhar, S., Lu, C.-T., Zhang, P.: A unified approach to detecting spatial outliers. GeoInformatica **7**(2):139–166 (2003)

Shekhar, S., Lu, C., Zhang, P.: Detecting graph-based spatial outliers: algorithms and applications (a summary of results). In: 7th ACM international conference on Knowledge discovery and data mining, pp. 371–376 (2001)

Shekhar, S., Ma, X.: GIS subsystem for a new approach to accessing road user charges

Shekhar, S., Schrater, P., Raju, W., Wu, W.: Spatial contextual classification and prediction models for mining geospatial data. IEEE Transactions on Multimedia (2002)

Shekhar, S., Vatsavai, R., Chawla, S., Burke, T.E.: Spatial Pictogram Enhanced Conceptual Data Models and Their Translation to Logical Data Models, Integrated Spatial Databases, Digital Maps, and GIS, Lecture Notes in Computer Science, vol. 1737 (1999)

Shekhar, S., Vatsavai, R., Ma, X., Yoo, J.: Navigation Systems: A Spatial Database Perspective. In: Schiller, J. and Voisard, A. (eds.) Location-Based Services, Chapter 3. Morgan Kaufmann, San Francisco, CA, USA, April (2004) ISBN-13: 978-1558609297

Shekhar, S., Vatsavai, R.R., Sahay, N., Burk, T.E., Lime, S.: Wms and gml based interoperable web mapping system. In ACM-GIS, pp. 106–111 (2001)

Shekhar, S., Yoo, J.S.: Processing In-Route Nearest Neighbor Queries: A Comparison of Alternative Approaches. ACM GIS (2003)

Shekhar, S., Zhang, P., Huang, Y., Vatsavai, R.R.: Trends in Spatial Data Mining. In: Kargupta, H., Joshi, A., Sivakumar, K., Yesha, Y. (eds.) Data Mining: Next Generation Challenges and Future Directions. AAAI/MIT Press, Cambridge, MA (2003)

Shengru, T., Abdelguerfi, M.: Web services for geographic information systems., IEEE Internet Comput. **10**, 13–15 (2006)

Shepard, D.: A two-dimensional interpolation function for irregularly spaced data. In: 23nd National Conference ACM, pp. 517–524. ACM (1968)

Sheppard, E., McMaster, R.B.: Scale and Geographic Inquiry: Nature, Society, and Method. Blackwell, Oxford (2004)

Sheppard, E., McMaster, R.B.: Scale and Geographic Inquiry: Nature Society and Method. Blackwell Publishing, Malden, MA (2004)

Sherali, H.D., Carter, T.B., Hobeika, A.G.: A location-allocation model and algorithm for evacuation planning under hurricane/flood conditions. Transp. Res. Part B, 439–452 (1991)

Sherman et al. (2006) QGIS user guide. http://qgis.org/releases/userguide.pdf. Accessed 21 Dec 2006

Sheth, A.: Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics. In: Goodchild, M., Egenhofer, M., Fegeas, R., Kottman, C. (eds.) Interoperating Geographic Information Systems, pp. 5–30. Kluver, New York (1999)

Shiffer, M.J.: Multimedia GIS for planning support and public disclosure. Cartography Geogr. Inf. Syst. **25**, 89–94 (1998)

Shim, R.: Wal-Mart to throw its weight behind RFID, http://news.com.com/2100-1022-1013767.html?tag=nl, CNET News: Retreived 10/10/2006 (2003)

Shi, W., Fisher, P.F., Goodchild, M.F. (eds.): Spatial Data Quality. Taylor & Francis, London (2002)

Shi, W., Liu, W.B.: A stochastic process-based model for the positional error of line segments in GIS. Int. J. Geogr. Inf. Sci. **14**, 51–66 (2000)

Shneiderman, B.: Designing the User Interface, Strategies for Effective Human-Computer Interaction, 3rd edn. Addison-Wesley Longman, Inc., Reading, Massachusetts (1998)

Shoichet, B.K., Bodian, D.L., Kuntz, I.D.: Molecular docking using shape descriptors. Journal of Computational Chemistry **13**(3), 380–397 (1992)

Shon, A.P., Grochow, K., Rao, R.P.N.: Robotic Imitation from Human Motion Capture using Gaussian Processes. In: Technical Report, Department of Computer Science and Engineering, University of Washington, Seattle, WA (2006)

Shoshani, A.: OLAP and statistical databases: Similarities and differences. In: Proc. 16th ACM Symp. Principles of Database Systems, pp. 185–196. Tucson, AZ, May (1997)

Shrinivasan, Y.B., Kraak, M.J.: Visualization of spatio-temporal patterns of public transport system data. In: Proceedings of the 22nd International Cartographic Conference, A Coruna Spain, 9–16 July 2005

Shu, N.C., Housel, B.C., Lum, V.Y.: CONVERT: A high level translation definition language for data conversion. Commun. ACM **18**, 557–567 (1975)

Sieber, R., Huber, S.: Atlas of Switzerland 2 – A highly interactive thematic national atlas. In: Cartwright, W., Peterson, M.P., Gartner, G.(eds.) Multimedia Cartography. 2nd edn., pp. 161–182. Springer, Berlin Heidelberg New York (2007)

Siegel, A.W., White, S.H., The development of spatial representations of large-scale environments. In: Reese, H.W. (ed.) Advances in Child Development and Behavior, pp. 9–55. Academic Press, New York (1975)

Siekierska, E.M., Taylor, F.D.R.: Electronic Mapping and Electronic Atlases: New Cartographic Products for the Information Era – The Electronic Atlas of Canada, CSIM Journal ACSGC, **45**(1), 11–21 (1991)

Siekierska, E., Williams, D.: National atlas of Canada on the internet and schoolnet. In: Köbben, B., Ormeling, F., Trainor, T. (eds.) Seminar on Electronic Atlases II, pp. 19–23. ICA Commission on National and Regional Atlases, Prague (1996)

Sierpinski, W.: Sur une nouvelle courbe qui remplit toute une aire plaine. Bull. Acad. Sci. Cracovie, SerieA 462–478 (1912)

Sigaev, T., Bartunov, O. (2006) GiST for PostgreSQL. http://www.sai.msu.su/~megera/postgres/gist. Accessed 1 Jan 2007

Silberschatz, A., Korth, H., Sudarshan, S.: Database System Concepts, 5th edn. McGraw-Hill, New York (2006)

Simmons, P.: Parts: A Study in Ontology. Clarendon, Oxford (1987)

Simon, H.: The New Science of Management Decisions. Harper and Row, New York (1960)

Simon, H.A.: The Sciences of the Artificial. MIT Press, Cambridge, MA (1996)

Simonovic, S.P., Bender, M.J.: Collaborative planning-support system: An approach for determining evaluation criteria. Journal of Hydrology **177**(3–4), 237–251 (1996)

Singh, A.: Digital change detection techniques using remotely-sensed data. Int. J. Remote Sens. **10**(6), 989–1003 (1989)

Singh, R., Macchi, L., Regazzoni, C., Plataniotis, K.: A statistical modelling based location determination method using fusion in WLAN. In: Proceedings of the International Workshop on Wireless Ad-hoc Networks, London, UK. (2005)

Singh, S., Mayfield, C., Cheng, R., Prabhakar, S.: The ORION database (http://www.orion.cs.purdue.edu).

Singh, V.P., Fiorentino M.: Geographical Information Systems in Hydrology, Water Science and Technology Library, vol. 26, ISBN 0-7923-4226-7. Kluwer Academic Publishers, Norwell, Massachusetts (1996)

Sinuany-Stern, Z., Stern, E.: Simulating the evacuation of a small city: the effects of traffic factors. Socio-Economic Planning Sciences, June 1993. Elsevier (1993)

Sistla, P., Wolfson, O., Chamberlain, S., Dao, S.: Modeling and Querying Moving Objects. IEEE ICDE, Birmingham, UK, 7–11 Apr 1997

Sistla, P.A., Wolfson, O., Chamberlain, S., Dao, S.: Querying the uncertain position of moving objects. In: Temporal Databases: Research and Practice. (1998)

Site Selection: New Advancements in Methods and Technology, Buckner, Robert W., National Book Network (1998)

Sithole, G., Vosselman, G.: Experimental comparison of filter algorithms for bare-earth extraction from airborne laser scanning point clouds. ISPRS J. Photogramm. Remote Sensing **59**, 85–101 (2004)

Slezak, D., Wang, G., Szczuka, M.S., Düntsch, I., Yao, Y. (eds.) Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, 10th International Conference RSFDGrC 2005 (Part I). Lecture Notes in Computer Science, vol. 3641, Aug 31–Sept 3, Springer, Regina, Canada (2005)

Slocum, T. A., Montello, D.R., Fuhrmann, S., Hedley, N.R., Blok, C., Jiang, B., Koussoulakou, A.: Cognitive and usability issues in geovisualization. Cartogr. Geogr. Inf. Sci. **28**, 61–75 (2001)

Smith, B.: Formal ontology, common sense and cognitive science. Int. J. Hum.-Comput. Stud. **43**, 641–667 (1995)

Smith, B., Mark, D.: Ontology and geographic kinds. In: Proceedings of the 8th International Symposium on Spatial Data Handling, Vancouver, BC, Canada, 12–15 July 1998

Smith, D., Lucey, B., Waller, L.A., Childs, J.E., Real, L.A.: Predicting the spatial dynamics of rabies epidemics on heterogeneous landscapes. Proceedings of the National Academy of Sciences, USA. **99**, 3668–3672 (2002)

Smith, D.A., Mann G.S.: Bootstrapping toponym classifiers. In: Proceedings of the Workshop on the Analysis of Geographic References, at the NAACL 2003 Conference (2003)

Smith, M.K.: Kurt Lewin, Groups, Experiental Learning and Action Research. The Encyclopedia of Informal Education. http://www.infed.org/thinkers/et-lewin.html (2001)

Snow, J.: On the Mode of Communication of Cholera. Churchill Livingstone, London, 2nd edn. (1854)

Snyder, J.P.: Flattening the Earth: Two Thousand Years of Map Projections. University of Chicago Press, Chicago (1993)

Soheili, A., Kalogeraki, V., Gunopulos, D.: Spatial Queries in Sensor Networks. Paper presented at the 13th International Symposium on Advances in Geographic Information Systems (GIS 2005), Bremen, Germany, November 2005

Sokal, R.R., Sneath, P.H.A.: Principles of numerical taxonomy. Freeman W.H., San Francisco, xvi, 359 p (1963)

Sonebraker, M., Rowe, L.: The Design of PostGRES, vol. 1. Association of Computing Machinery, Berkeley, CA, USA (1986)

Song, S.I., Kim, Y.H., Yoo, J.S.: An Enhanced Concurrency Control Scheme for Multidimensional Index Structure. In: IEEE Transactions on Knowledge and Data Engineering No. 1, vol. 16, pp. 97–111. January 2004

Song, Z., Roussopoulos, N.: K-Nearest Neighbor Search for Moving Query Point. SSTD, Redondo Beach, CA, USA, 12–15 July 2001

Sorensen, J., Vogt, B.: Interactive Emergency Evacuation Planning Guidebook. Chemical Stockpile Emergency Preparedness Program, Department of Homeland Security. [Available online at: http://emc.ornl.gov/CSEPPweb/evac_files/index.htm] (2006)

Sorensen, M., Sayegh, F.: The initiation, growth, sustainment of SDI in the Middle East—notes from the trenches. Inf. Technol. Dev. **13**, 95–100 (2007)

Sorrows, M.E., Hirtle, S.C.: The Nature of Landmarks for Real and Electronic Spaces. In: Freksa, C., Mark, D. (eds.) Spatial Information Theory, pp. 37–50. Springer, Berlin (1999)

Sousa, M.C., Prusinkiewicz, P.: A few good lines: suggestive drawing of 3D models. In: Proceedings of Eurographics, Computer Graphics Forum, 22(3):381–390 (2003)

Sowa, J.F., Zachman, J.A.: Extending and Formalizing the Framework for Information Systems Architecture. IBM Syst. J. **31**(3). IBM Publication G321–5488 (1992)

Space Imaging Middle East (2004) http://www.spaceimagingme.com/content/faq/Questions/index.asp. Cited: March 23, 2007

Spanish Minister of Interior: Framework for Action 2005–2015 (2005)

Spatial Autocorrelation, Natural resources Canada. http://www.pfc.forestry.ca/profiles/wulder/mvstats/spatial_e.html (2006). Accessed 10 Apr 2006

Spatial Data Infrastructure Cookbook, http://www.gsdi.org/gsdicookbookindex.asp This SDI Implementation Guide or Cookbook, through the support of the Global Spatial Data Infrastructure community, provides geographic information providers and users with the necessary background information to evaluate and implement existing components of SDI. It also facilitates participation within a growing (digital) geographic information community known as the Global Spatial Data Infrastructure (GSDI)

Spence, R., Information Visualization. Harlow: Addison Wesley/ACM (2001)

Spiegelhalter, D.J., Best, N.G., Carlin, B.P., van der Linde, A.: Bayesian measures of model complexity and fit (with discussion and rejoinder). J. R. Statistical Soc. Series B, **64**, 583–639 (2002)

Spiegelhalter, D., Thomas, A., Best, N., Gilks, W.: Bugs.5 Bayesian inference using Gibbs sampling. Manual, version ii. MRC Biostatistics Unit, Institute of Public Health, Cambridge, UK (1996)

Spiess, E.: The need for generalization in a gis environment. In: Müller, J.C., Lagrange, J.P., Weibel, R. (eds.) GIS and Generalization – Methodology and Practice, pp. 31–46. Taylor & Francis, London (1995)

Sprague, R.H.: Framework for DSS. MIS Quart. **4**(4), 1–26 (1980)

Sproull, R.: Refinements to nearest neighbor searching in K-dimensional trees. Algorithmica **6**, 579–589 (1991)

SQL Server Magazine: City of Portland Tames Massive SQL Server Data Warehouse. June 27. Available at www.sqlmag.com (2002)

Srinivasan, R., Arnold, J.G.: Integration of a Basin-Scale Water Quality Model with GIS. Water Resour. Bull. **30**:453–462 (1994)

Srinivasan, R., Ramanarayanan, R., Arnold, J., Bednarz, S.: Large area hydrologic modeling and assessment part II: Model application. J. Am. Water Resour. Assoc. **34**(1), 91–101 (1998)

Srivastava, L.: Ubiquitous network societies: the case of radio frequency identification. Paper presented at: the ITU workshop on ubiquitous network societies, International Telecommunication Union, April 6–8 2005

Staab, S., Studer, R.: Handbook on Ontologies. Springer-Verlag, Heidelberg (2004)

Stanoi, I., Riedewald, M., Agrawal, D., Abbadi, A.E.: Discovery of Influence Sets in Frequently Updated Databases. In: Proceedings of the 27th International Conference on Very Large Data Bases (VLDB'01) pp. 99–108. Morgan Kaufmann Publishers Inc. (2001)

Stassopoulou, A., Caelli, T.: Building detection using Bayesian networks. Int. J. Pattern Recognit. Artif. Intell. **14**(6), 715–733 (2000)

Stassopoulou, A., Petrou, M., Kittler, J.: Application of a Bayesian network in a GIS based decision making system. Int. J. Geograph. Info. Sci. **12**(1), 23–45 (1998)

A status report on location analysis – retail location analysis – Guest Column – Column, DSN Retailing Today, Jan 26, 2004 by David Rogers http://www.findarticles.com/p/articles/mi_m0FNP/is_2_43/ai_112792451

Stefanidis, A., Nittel, S.: GeoSensor Networks. CRC, Boca Raton, FL (2004)

Stefanovic, N., Han, J., Koperski, K.: Object-based selective materialization for efficient implementation of spatial data cubes. IEEE Trans. Knowl. Data Eng. **12**, 938–958 (2000)

Steinhaus, H.: Mathematical Snapshots, 3rd edn. Dover, New York (1999), pp. 256–259

Steiniger, S., Neun, M., Edwardes, A.: Foundations of Location Based Services (2006). Available via http://www.geo.unizh.ch/publications/cartouche/lbs_lecturenotes_steinigeretal2006.pdf

Stein, M.L.: Space-time covariance functions. J. Am. Stat. Assoc. **100**, 320–321 (2005)

Stell, J.G.: Granularity in change over time. In: Duckham, M., Goodchild, M., Worboys, M. (eds.) Foundations of Geographic Information Science. pp. 95–115. Taylor and Francis (2003)

Stell, J.G.: Part and complement: Fundamental concepts in spatial relations. Ann. Artif. Intell. Mat. **41**,1–18 (2004)

Stell, J.G.: The representation of discrete multi-resolution spatial knowledge. In: Cohn, A.G., Giunchiglia, F., Selman, B. (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference (KR2000), pp. 38–49. Morgan Kaufmann (2000)

Stell, J.G., Worboys, M.F.: Stratified map spaces: A formal basis for multi-resolution spatial databases. In: Poiker, T.K., Chrisman, N. (eds.) Proceedings 8th International Symposium on Spatial Data Handling SDH'98, pp. 180–189. International Geographical Union (1998)

Stevens, A., Coupe, P.: Distortions in Judged Spatial Relations 10. In: Cognitive Psychology **10**(4), 422–437 (1978)

Stockman, H.: Communication by means of reflected power, Proc. Inst. Radio Engin. **36**, 1196–1204 (1948)

Stojmenovic, I., Miyakawa, M.: An optimal parallel algorithm for solving the Maximal Elements Problem in the Plane. Parallel Computing (1988) **7**(2):249–251

Stokes, D.E.: Pasteur's Quadrant: Basic Science and Technological Innovation. Brookings Institution Press, Washington, DC (1997)

Stolze, K.: SQL/MM Spatial – The Standard to Manage Spatial Data in a Relational Database System. In: Datenbanksysteme für Business, Technologie und Web, Tagungsband der 10, pp. 247–264. Leipzig, Germany (2003)

Stoney, W.E.: Markets and Opportunities. Earth Imag. J. **2**(1), 10–14 (2005)

Stora, D., Agliati, P.O., Cani, M.P., Neyret, F., Gascuel, J.D.: Animating lava flows. In: Graphics Interface, (1999)

Straub, B.M., Wiedemann, C.: Towards the update of geodata by automatic object extraction. In: Serpico, S.B (ed.) Image and Signal Processing for Remote Sensing VI. SPIE Proceedings vol. 4170, p. 304. The International Society for Optical Engineering Bellingham, WA (2000) http://ipi216.ipi.uni-hannover.de/html/publikationen/2000/straub/straub_gerke_wiedemann.pdf

Strebelle, S.: Conditional simulation of complex geological structures using multiple-point statistics. Math. Geo. **34**, 1–21 (2002)

Strothotte, T., Schlechtweg, S.: Non-Photorealistic Computer Graphics: Modeling, Rendering and Animation. Morgan Kaufman, San Francisco (2002)

Stubkjær, E.: Modelling units of real property rights. In: Virrantaus, K., Tveite, H. (eds.) ScanGIS'03 Proceedings, 9th Scandinavian Research Conference on Geographical Information Sciences, June 2003. Espoo, Finland (2003)

Submit Corner (2001) 8 Second Rule. http://www.submitcorner.com/Guide/Bandwidth/001.shtml. Accessed 29 Aug 2007

Sung, K., Bell, M.G.H., Seong, M., Park, S.: Shortest paths in a network with time-dependent flow speeds. Eur J Oper Res **121**(1), 32–39 (2000)

Sun, H., Cornish, P.S.: Estimating shallow groundwater recharge in the headwaters of the Liverpool Plains using SWAT. Hydrol. Process. **19**(3), 795–807 (2005)

Sun P., Chawla, S.: On local spatial outliers. In: 4th IEEE International Conference on Data Mining, pp. 209–216 (2004)

Suwanwerakamtorn, R.: GIS and hydrologic modeling for the management of small watersheds. Intern. J. Appl. Earth Obs. Geoinf. (ITC) 1994–4, 343–348 (1994)

SVG Community: SVG.Org – an SVG community page, news and links for SVG (2004–2006), http://www.svg.org/, accessed 2006–11–12

Swaminathan, R., Grossberg, M.D., Nayar, S.K.: A perspective on distortions. Proc. Comput. Vision Pattern Recog. 2:594–601 (2003)

Swan, D., Aggett, G.: Two-dimensional modeling of stream channel dynamics in the Naches River, Washington. Transactions in GIS, in review (2007)

Sweeney, L.: Achieving *k*-anonymity Privacy Protection using Generalization and Suppression. Int. J. Uncertain. Fuzziness Knowl. Sys. **10**(5), 571–588 (2002)

Sweeney, L.: *k*-anonymity: A Model for Protecting Privacy. Intern. J. Uncertain., Fuzziness Knowl.-based Syst. **10**(5), 557–570 (2002)

Takeuchi, H., Schmidt, A.H.: The new promise of computer graphics. Harvard Bus. Rev. **58**(1), 122–131 (1980)

Tango, T.: Comparison of general tests for spatial clustering. In: Lawson, A.B., Biggeri, A., Boehning, D., Lesaffre, E., Viel, J.F., Bertollini, R. (eds.) Disease Mapping and Risk Assessment for Public Health, chapter 8. Wiley, New York (1999)

Tango T., Takahashi, K.: flexscan: Software for the flexible scan statistics, version 1.1.2. http://www.niph.go.jp/soshiki/gijutsu/download/exscan/index.html (2005)

Tan, K.L., Eng, P.K., Ooi, B.C.: Efficient Progressive Skyline Computation. In: Proceedings of VLDB'01, pp. 301–310 (2001)

Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to data mining. Addison-Wesley, Upper Saddle River (2005)

Tao, C.: Online GIServices. Journal of Geospatial Engineering. **3**(2), 135–143 (2001)

Tao, V.: Microsoft Virtual Earth: New horizons in on-line mapping. Paper presented at GeoInformatics'2006, Wuhan, China, October 28–29 (2006)

Tao, Y., Cheng, R., Xiao, X., Ngai, W.K., Kao, B., Prabhakar, S.: Indexing multi-dimensional uncertain data with arbitrary probability density functions. In: Proc. VLDB, 2005

Tao, Y., Faloutsos, C., Papadias, D., Liu, B.: Prediction and Indexing of Moving Objects with Unknown Motion Patterns. In: Proc. of ACM Management of Data (SIGMOD) (2004)

Tao, Y., Papadias, D.: MV3R-Tree: A spatio-temporal access method for timestamp and interval queries. In: Proc. of Very Large Data Bases (VLDB), pp. 431–440 (2001)

Tao, Y., Papadias, D., Shen, Q.: Continuous nearest neighbor search. In: Proc. VLDB, 2002

Tao, Y., Papadias, D.: Spatial Queries in Dynamic Environments. ACM Trans. Database Syst. **28**(2), 101–139 (2003)

Tao, Y., Papadias, D., Sun, J.: The TPR*-Tree: An Optimized Spatio-temporal Access Method for Predictive Queries. In: Proceedings of the VLDB Conference, pp. 790–801. Berlin, Germany, 9–12 Sept 2003

Tao, Y., Papadias, D.: Time-Parameterized Queries in Spatio-temporal Databases. ACM SIGMOD, Wisconsin, 3–6 June 2002

Tao, Y., Sun, J., Papadias, D.: Analysis of predictive spatio-temporal queries. ACM Trans Database Syst **28**(4), 295–336 (2003)

Tao, Y., Zhang, J., Papadias, D., Mamoulis, N.: An efficient cost model for optimization of nearest neighbor search in low and medium dimensional spaces. IEEE TKDE **16**(10):1169–1184 (2004)

Tarantilis, C.D., Gendreau, M., Spinellis, D.: The Impact of Metaheuristic Strategies in Transportation and Logistics Applications. IEEE Intelligent Syst. **20**(4), 16–18 (2005)

Tarantilis, C.D., Kiranoudis, C.T.: Using a spatial decision support system for solving the vehicle routing problem. Inf. Manag. **39**(5), 359–375 (2002)

Tarantilis, C.D., Kiranoudis, C.T.: Using the Vehicle Routing Problem for the Transportation of Hazardous Materials. Oper. Res. An. Int. J. **1**(1), 67–78 (2001)

Tarantilis, C.D., Kiranoudis, C.T., Vassiliadis, V.S.: A backtracking adaptive threshold accepting metaheuristic method for the Vehicle Routing Problem. Syst. Anal. Model. Simul. **42**(5), 631–644 (2002)

Tarantilis, C.D., Kiranoudis, C.T., Vassiliadis, V.S.: A list based threshold accepting algorithm for the capacitated vehicle routing problem. Int. J. Comput. Math. **79**(5), 537–553 (2002)

Tarski, A.: A Decision Method for Elementary Algebra and Geometry. University of California Press, Berkeley (1948)

Tate, E., Maidment, D., Olivera, F., Anderson, D.: Creating a Terrain Model for Floodplain Mapping, ASCE. J. Hydrol. Eng. **7**(2), 100–108 (2002)

Tate, N.J., Atkinson, P.T.: Modelling Scale in Geographic Information Science. Wiley, Chichester (2001)

Taubman, D.: High performance scalable image compression with EBCOT. IEEE Trans. Image Processing **9**, 1158–1170 (2000)

Tayeb, J.: Design and Performance Evaluation of Indexing Methods for Dynamic Attributes in Mobile Database Management Systems. Master's thesis, Bilkent University, May 1997, Ankara, Turkey (1997)

Tayeb, J., Ulusoy, O., Wolfson, O.: A Quadtree Based Dynamic Attribute Indexing Method. The Computer Journal **41**(3), 185–200 (1998)

Taylor, K.J.: Bayesian Belief Networks: A Conceptual Approach to Assessing Risk to Habitat. Utah State University, Logan, UT (2003)

Taylor, P.J.: Is there a Europe of cities? World cities and the limitations of Geographical Scale Analyses. In: Sheppard, E., McMaster, B. (eds.) Scale and Geographic Inquiry: Nature, Society, and Method, pp. 213–235 Blackwell Publishing, Malden, MA (2004)

Taylor, P.J.: Quantitative Methods in Geography: An Introduction to Spatial Analysis. Houghton Mifflin, Boston (1977)

Teh, C.-H., Chin, R.T.: On image analysis by the methods of moments. IEEE Trans. on Pattern. Anal. Mach. Intell. **10**(4), 496–513 (1988)

Tenmoku, R., Kanbara, M., Yokoya, N.: A wearable augmented reality system using positioning infrastructures and a pedometer, Wearable Computers, 2003. Proceedings. Seventh IEEE International Symposium on 110–117 (2003)

TerraSIP. Terrasip introduction. http://terrasip.gis.umn.edu/

Terrovitis, M., Bakiras, S., Papadias, D., Mouratidis, K.: Constrained Shortest Path Computation. In: Proceedings of the 9th International Symposium on Spatial and Temporal Databases (SSTD'05), volume 3633 of Lecture Notes in Computer Science, pages 181–199. Springer, August (2005)

Thayer, W.C., Griffith, D.A., Goodrum, P.E., Diamond, G.L., Hassett, J.M.: Application of Geostatistics to Risk Assessment. Risk Analysis: An Int. J. **23**(5), 945–960 (2003)

Theme issue "Airborne Laser Scanning": ISPRS J. Photogramm. Remote Sensing **54**

Theodoridis, Y.: The R-tree-Portal. (2003)

Theodoridis, Y., Sellis, T.: A model for the prediction of r-tree performance. In: Proc. of ACM Symposium on Principles of Database Systems (PODS), pp. 161–171 (1996)

Theodoridis, Y., Sellis, T., Papadopoulos, A., Manolopoulos, Y.: Specifications for efficient indexing in spatiotemporal databases. In: SSDBM, pp. 123–132 (1998)

Theodoulou, G.: Contraflow Evacuation on the Westbound I-10 Out of the City of New Orleans. Master's Thesis, Department of Civil and Environmental Engineering, Louisiana State University (2003)

Theodoulou, G., Wolshon, B.: Alternative methods to increase the effectiveness of freeway contraflow evacuation. J. Trans. Res. Board. Transportation Research Record 1865, 48–56 (2004)

Theodoulou, G., Wolshon, B.: Modeling and Analyses of Freeway Contraflow to Improve Future Evacuations. Presented at The 83rd Annual Meeting of Transportation Research Board, Washington, D.C. 2004

Thibos, L.N., Applegate, R.A., Schwiegerling, J.T., Webb, R.: Standards for reporting the optical aberrations of eyes. In: TOPS, Santa Fe, NM, USA (1999)

Thiebaux, H.J., Pedder, M.A.: Spatial Objective Analysis with Applications in Atmospheric Science. Academic Press (1987)

Thill, J.-C.: Geographic information systems in transportation research. Elsevier Science, Oxford, England (2000)

Thompson, R.B.: Global Positioning System: The Mathematics of GPS Receivers. Math. Magazine **71**(4), 260–269 (1998)

Ticheler, J.: SDI Architecture diagram. http://193.43.36.138/relatedstuff/index_html/document_view Accessed 10 Mar 2006

Tiefelsdorf, M.: Modelling Spatial Processes – The Identification and Analysis of Spatial Relationships in Regression Residuals by Means of Moran's I. Lecture Notes in Earth Sciences, vol. 87. Springer-Verlag, Berlin (2000)

Timmermans, H., Arentze, T., Joh, C.-H.: Analyzing space-time behaviour: New approaches to old problems. Prog. Hum. Geogr. **26**, 175–190 (2002)

Timpf, S.: Abstraction, levels of detail, and hierarchies in map series. In: Freksa, C., Mark, D.M. (eds.) Spatial Information Theory – cognitive and computational foundations of geographic information science (COSIT'99). Lecture Notes in Computer Science 1661, pp. 125–140. Springer-Verlag, Berlin-Heidelberg (1999)

Timpf, S., Volta, G.S., Pollock, D.W., Egenhofer, M.J.: A Conceptual Model of Wayfinding Using Multiple Levels of Abstractions. In: Frank, A.U., Campari, I., Formentini, U. (eds.) Theories and Methods of Spatio-temporal Reasoning in Geographic Space. Lecture Notes in Computer Science 639, pp. 348–367. Springer Verlag, Heidelberg-Berlin (1992)

Tobler, W.: Cellular geography. In: Philosophy in Geography, pages 379–386, Dordrecht Reidel Publishing Company, Dordrecht, Holland (1979)

Tobler, W. R.: A computer movie simulating urban growth in the Detroit region. Econ. Geogr. **46**, 234–240 (1970)

Tobler, W.: Smooth pycnophylactic interpolation for geographical regions. J. Am. Stat. Assoc. **74**, 519–530 (1979)

Tolle, G.,Polastre, J., Szewczyk, R., Culler, D., Turner, N., Tu, K., Burgess, S., Dawson, T., Buonadonna, P., Gay, D., Hong, W.: 2005. A Macroscope in the Redwoods. International Conference on Embedded Networked Sensor Systems (ACMSensys), San Diego, November (2005)

Tomlin, C.D.: Geographic Information Systems and Cartographic Modeling. Prentice Hall, Englewood Cliffs, NJ (1990)

Tomlinson, R.: Thinking about GIS: geographic information system planning for managers. ESRI Press, Redlands, CA (2003)

TomTom, http://www.tomtom.com/ (2007)

Topaloglou, T.: First order theories of approximate space. In: Anger, F. et al. (ed.): Working notes of AAAI workshop on spatial and temporal reasoning, Seattle, pp. 283–296 (1994)

Torrens, P.M.: Simulating sprawl. Ann. Assoc. Am. Geogr. **96**, 248–275 (2006)

Toth, P., Vigo, D.: The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia (2002)

Tøssebro, E., Nygård, M.: An advanced discrete model for uncertain spatial data. In: Meng, X., Su, J., Wang, Y. (eds.) Advances in Web-Age Information Management, Third International Conference (WAIM'02). Lecture Notes in Computer Science, vol. 2419, pp. 37–51. Springer, Berlin (2002)

Trajcevski, G.: Probabilistic range queries in moving objects databases with uncertainty. In: MobiDE (2003)

Trajcevski, G., Scheuermann, P., Ghica, O., Hinze, A., Voisard, A.: Evolving triggers for dynamic environments. In: Extending Database Technology (EDBT) (2006)

Trajcevski, G., Wolfson, O., Hinrichs, K., Chamberlain, S.: Managing uncertainty in moving objects databases. ACM Transactions on Database Systems. **29**(3) (2004)

Traveling Salesman Problem. History of the TSP. Jan. (2005)

Traveling Salesman Problem website supported by School of Industrial and Systems Engineering at Georgia Tech. http://www.tsp.gatech.edu/

Trolltech (2006) Qt documentation. http://doc.trolltech.com/. Accessed 21 Dec 2006

Tryfona, N., Jensen, C.S.: Conceptual Data Modeling for Spatiotemporal Application. GeoInformatica **3**(3), 245-268 (1999)

Tryfona, N., Jensen, C.S.: Using Abstractions for Spatio-temporal Conceptual Modeling. In: Proceedings of the 14th Annual Symposium on Applied Computing (SAC'00). Villa Olmo, Como, Italy (2000)

Tryfona, N., Price, R., Jensen, C.S.: Conceptual Models for Spatio-Temporal Applications. In Spatiotemporal Databases: The CHOROCHRONOS approach, Lecture Notes in Computer Science, vol. 2520, Chapter 3. Springer-Verlag, Berlin/Heidelberg (2003)

Tryfona, N., Price, R., Jensen, C.S.: Spatiotemporal Conceptual Modeling.In: Spatiotemporal Databases: The CHOROCHRONOS approach. Springer-Verlag, Berlin (2003)

Tryon, R.C.: Cluster analysis; correlation profile and orthometric (factor) analysis for the isolation of unities in mind and personality., Ann Arbor, Mich.: Edwards brother, inc., lithoprinters and publishers. viii, 122 p (1939)

Tsou, M.H., Buttenfield, B.P.: An Agent-based, Global User Interface for Distributed Geographic Information Services. Proceedings 7th International Symposium on Spatial Data Handling, Vancouver, British Columbia, July, 1998. 603–612 (1998)

Tsou, M.H., Buttenfield, B.P.: A Dynamic Architecture for Distributed Geographic Information Services. Transactions in GIS. **6**(4), 355–381 (2002)

Tsou, M.H.: An Operational Metadata Framework for Searching, Indexing, and Retrieving Distributed Geographic Information Services on the Internet. In Geographic Information Science (GIScience 2002), Egenhofer, M., Mark, D. (eds.), Lecture Notes in Computer Science vol 2478, pp. 313–332. Springer-Verlag, Berlin (2002)

TSP Home Web Site. http://www.tsp.gatech.edu/

Tucker, K., Rushton, S.P., Sanderson, R.A., Martin, E.B., Blaiklock, J.: Modelling bird distributions – a combined GIS and bayesian rule-based approach. Landsc. Ecol. **12**(2), 77–93 (1997)

Tufekci, S.: An integrated emergency management decision support system for hurricane emergencies. University of Florida, Safety Science, July 1995 (1995)

Tufekci, S., Kisko, T.M.: Regional Evacuation Modeling System (REMA): A decision support system for emergency area evacuation. Computers & Industrial Engineering. Elsevier (1991)

Tunkelang, D.: Making the Nearest Neighbor Meaningful. SIAM WOrkship on Clustering High Dimensional Data and its Applications. (2002)

Turdukulov, U.D., Kraak, M.J.: Designing a visual environment for exploration of time series of remote sensing data: in search for convective clouds. Comput. Graphics **31**, 370–379 (2007)

Tuydes, H., Ziliaskopoulos, A.: Network re-design to optimize evacuation contraflow. Technical Report 04-4715, Presented at 83rd Annual Meeting of the Transportation Research Board (2004)

Tversky, A.: Features of similarity. Psychol. Rev. **84**(4), 327–352 (1977)

Twa, M.D., Parthasarathy, S., Raasch, T.W., Bullimore, M.A.: Automated classification of keratoconus: A case study in analyzing clinical data. In: SIAM Int'l Conference on Data Mining, San Francisco, CA, 1–3 May 2003

Tyler, M.: Web Mapping Illustrated – Using Open Source GIS Toolkits, O'Reilly, Sebastopol CA (2003) ISBN 0-596-00865-1, 349 pages

Tzouramanis, T., Vassilakopoulos, M., Manolopoulos, Y.: Overlapping linear quadtrees and spatio-temporal query processing. Comput. J. **43**(3), 325–343 (2000)

Ullman, J.D.: Information integration using logical views. In: Proceedings of the International Conference on Database Theory (ICDT) 1997, January 8–10, Delphi, Greece

Ullman, J.D.: Principles of Database and Knowledge-Base Systems. Computer Science Press, New York (1989)

Understanding Raster Data And Cell-Based Modeling. http://www.cfr.washington.edu/research.urbaneco/GIS_Tutorial Accessed 21 February 2007

United States Army Corps of Engineers: Hydrologic modeling system HEC-HMS: User's manual. USACE Hydrologic Engineering Center, Washington, DC (2001)

United States Army Corps of Engineers: Southeast United States Hurricane Evacuation Traffic Study, performed by Post. Buckeley Schuh and Jernigan Inc., Tallahassee, FL (2000)

United States Geologic Survey, Content Metadata Standards for Marine Science: A Case Study, USGS Open-File Report 2004-1002. http://pubs.usgs.gov/of/2004/1002/html/evol.html

United States Geologic Survey, Formal metadata: information and software. http://geology.usgs.gov/tools/metadata/ (2006)

The United States Navy: Identification Friend or Foe. http://www.nrl.navy.mil/content.php?P=FRIENDFOE, United States Naval Research Library: Retreived 11/05/2006 (2006)

United States Nuclear Regulatory Commission: The Accident At Three Mile Island Fact Sheet. United States Nuclear Regulatory Commission, Washington, DC (2000)

University of California (Berkeley) (1999) The BSD License. http://www.opensource.org/licenses/bsd-license.php. Accessed 1 Jan 2007

University of Minnesota. MapServer, MapServer Website: http://mapserver.gis.umn.edu/ (2006)

Urbanik, T., Desrosiers, A.E.: An Analysis of Evacuation Time Estimates around 52 Nuclear Power Plant Sites. US Nuclear Regulatory Commission (1981)

Urbanke, S.: AdV – Projekt "ATKIS – Generalisierung" 53. Deutscher Kartographentag, Rostock, 22 Sep 2005 (in German)

Urbanke, S., Dieckhoff, K.: The adv-project atkis generalization, part model generalization (in German). Kartographische Nachrichten **56**(4), 191–196 (2006)

Urbina, E., Wolshon, B.: National Review of Hurricane Evacuation Plans and Policies: A Comparison and Contrast of State Practices. Transp. Res., Part A: Policy Pract. **37**(3), 257–275 (2003)

US Census Bureau: Census 2000 PHC-T-40. Estimated Daytime Population and Employment-Residence Ratios: 2000; Technical Notes on the Estimated Daytime Population. http://www.census.gov/population/www/socdemo/daytime/daytimepoptechnotes.html (2000). Accessed 17 Sept 2007

US Census Bureau. TIGER. http://tiger.census.gov/.

US Census Bureau: US Census Bureau's Fact Finder for the Nation: History and Organization. http://www.census.gov/prod/2000pubs/cff-4.pdf (2000). Accessed 17 Sept 2007

U.S. Department of Homeland Security: National Incident Management System. US Department of Homeland Security, Washington, D.C. (2004)

U.S. Department of Homeland Security: Target Capabilities List: Version 1.1. US Department of Homeland Security, Office of State and Local Government Coordination and Preparedness, Washington, DC (2005)

US Department of Transportation, US Department of Homeland Security: Report to Congress on Catastrophic Hurricane Evacuation Plan Evaluation, June 1 (2006)

U.S. EPA: National Center for Environmental Assessment. http://www.epa.gov/nheerl/wtc/index.html. WTC (World Trade Center) Report Link: Exposure and Human Health Evaluation of Airborne Pollution from the World Trade Center Disaster (External Review Draft). WTC Report: Toxicological Effects of Fine Particulate Matter Derived from the Destruction of the World Trade Center. (2002)

U.S. EPA: Projecting Land-Use Change: A Summary of Models for Assessing the Effects of Community Growth and Change on Land-Use Patterns. EPA/600/R-00/098. U.S. Environmental Protection Agency, Office of Research and Development, Cincinnati, OH (2000)

U.S. EPA: Strategic Plan for Homeland Security. http://www.epa.gov/epahome/downloads/epa_homeland_security_strategic_plan.pdf (2002)

US Federal Communications Commission: Revision of Part 15 of the Commission's Rules Regarding Ultra-Wideband Transmission Systems, FCC 02–48, First Report & Order, April 2004

US Geological Survey. Spatial Data Transfer Standard Informational Website, http://mcmcweb.er.usgs.gov/sdts, as viewed November 26 (2006)

Using Dublin Core. http://www.dublincore.org/documents/2001/04/12/usageguide/. Accessed 13 Sept 2007

Vance, D., Eisenberg, R., Walsh, D.: Inside AutoCAD Map 2000, The Ultimate How-To Resource and Desktop Reference for AutoCAD Map. OnWord Press, Florence (2000)

Vance, D., Smith, C., Appell, J.: Inside Autodesk World. OnWord Press, Santa Fe, NM (1998)

Vance, D., Walsh, D., Eisenberg, R.: Inside AutoCAD Map 2000. Autodesk Press, San Rafael (2000)

Van den Bercken, V., Blohsfeld, B., Dittrich, J., Krämer, J., Schäfer, T., Schneider, M., Seeger, B.: XXL – a library approach to supporting efficient implementations of advanced database queries. In: VLDB, pp. 39–48 (2001)

Vandeurzen, L.: Logic-Based Query Languages for the Linear Constraint Database Model. PhD thesis, Limburgs Universitair Centrum (1999)

van Kreveld, M., Reinbacher, I., Arampatzis, A., van Zwol, R.: Distributed Ranking Methods for Geographic Information Retrieval. In: Developments in Spatial Data Handling, pp. 231–243. Springer, Berlin, Heidelberg (2004)

Van Liew, M.W., Garbrecht, J.D., Arnold, J.G.: Simulation of the impacts of flood retarding structures on streamflow for a watershed in southwestern Oklahoma under dry, average, and wet climatic conditions. J. Soil Water Conserv. **58**(6), 340–348 (2003)

Van Oijen, M., Rougier, J., Smith, R.: Bayesian calibration of process-based forest models: bridging the gap between models and data. Tree Physiol. **25**, 915–927 (2005)

van Oosterom, P.: The gap-tree, an approach to 'on-the-fly' map generalization of an area partitioning. [20], 120–132

van Oosterom, P., Schenkelaars, V.: The development of an interactive multiscale GIS. Int. J. Geogr. Inf. Syst. **9**, 489–507 (1995)

van Oosterom, P., Schlieder, C., Zevenbergen, J., Hess, C., Lemmen, C., Fendel, E. (eds.): Standardization in the cadastral domain. In: Proceedings, Standardization in the Cadastral Domain, Bamberg, December 2004 Frederiksberg. The International Federation of Surveyors, Germany (2005) van Oosterom, P.: A storage structure for a multiscale database: the reactive-tree. Comput., Environ. Urban Syst. **16**, 239–47 (1992)

van Oosterom, P., van den Bos, J.: An object-oriented approach to the design of geographic information systems. Comput. Graphics **13**, 409–418 (1989)

van Oosterom, P.: Variable-scale topological data structures suitable for progressive data transfer: The GAP-face tree and GAP-edge forest. Cartogr. Geogr. Inf. Sci. **32**, 331–346 (2005)

van Smaalen, J.: Automated Aggregation of Geographic Objects. A new Approach to the Conceptual Generalisation of Geographic Databases. PhD thesis, Wageningen University, The Netherlands (2003)

van Wijk, J.J., van Selow, E.: Cluster and calendar-based visualization of time series data. In: Proceedings 1999 IEEE Symposium on Information Visualization, 25–26 October 1999, pp 4–9. IEEE Computer Society

Varela, F.J., Thompson, E., Rosch, E.: The Embodied Mind. MIT Press, Cambridge Massachusetts (1997)

Varis, O., Kuikka, S.: An influence diagram approach to Baltic salmon management. In: Proceedings Conference on Decision Analysis for Public Policy in Europe. INFORMS Decision Analysis Society, Atlanta, GA (1996)

Varman, P.J., Verma, R.M.: An efficient multiversion access structure. TKDE **9**(3), 391–409 (1997)

Vasseur, B., Devillers, R., Jeansoulin, R. : Ontological Approach of the Fitness of Use of Geospatial Datasets. In: Proceedings of 6th AGILE Conference on Geographic Information Science, pp. 497–504. Lyon, France (2003)

Vatsavai, R.R., Burk, T.E., Wilson, B.T., Shekhar, S.: A web-based browsing and spatial analysis system for regional natural resource analysis and mapping. In: Proceedings of the eighth ACMGIS, Washington D.C., USA (2000)

Vatsavai, R.R., Shekhar, S., Burk, T.E., Lime, S.: Umn-mapserver: An high-performance, interoperable, and open source web mapping and geo-spatial analysis system. In: GIScience (2006)

Vdraft. Vdraft intenet tools. http://www.vdraft.com/vtools.html

Vega-Redondo, F.: Diffusion, Search and Play in Complex Social Networks, forthcoming. Econometric Society Monograph Series (2006)

Ventura, S., Miller, T., Barry, G.: Community engagement in land use planning through web-based technologies. In: Geertman, S., Stillwell, J. (eds.) Planning Support Systems in Practice, pp. 87–98. Springer, New York, NY (2003) See also web site for study at http://www.lic.wisc.edu/shapingdane/

Verhein, F., Chawla, S.: Mining spatio-temporal association rules, sources, sinks, stationary regions and thoroughfares in object mobility databases. In: Proceedings of the 11th International Conference on Database Systems for Advanced Applications. Lecture Notes in Computer Science, vol. 3882, pp. 187–201. Springer, Berlin, Heidelberg (2006)

Vernon Jr, F.: Application of the microwave homodyne, Antennas and Propogation, Transactions of the IRE Professional Group on **4**(1), 110–116 (1952)

Versavel, J.: Road safety through video detection. In: Proceedings of IEEE International Conference on Intelligent Transportation System, Boulder, Colorado, 1999, pp. 753–757

Viaña, R., Magillo, P., Puppo, E., P.A. Ramos: Multi-VMap: a multi-scale model for vector maps. GeoInformatica, **10**, 359–394 (2006)

Vidal-Rodeiro, C., Lawson, A.: Monitoring changes in spatio-temporal maps of disease. Biom. J. **48**, 1–18 (2006)

Vieux, B.E., Bedient P.B.: Assessing urban hydrologic prediction accuracy through event reconstruction, Special Issue on Urban Hydrology. J. Hydrol. **299**(3–4), 217–236 (2004)

Vieux, B.E.: DEM Aggregation and Smoothing Effects on Surface Runoff Modeling. J. Comput. Civ. Eng. **7**(3), 310–338 (1993)

Vieux, B.E.: Distributed Hydrologic Modeling Using GIS. Water Science Technology Series, vol. 48, ISBN 1-4020-2459-2, p. 289, CD-ROM including model software and documentation, 2nd edn. Kluwer Academic Publishers, Norwell, Massachusetts (2004)

Vieux, B.E., Moreda F.G.: Ordered Physics-Based Parameter Adjustment of a Distributed Model. In: Duan Q., Sorooshian S., Gupta H.V., Rousseau A.N., Turcotte, R. (eds.) Advances in Calibration of Watershed Models. Water Science and Application Series, vol. 6, pp. 267–281. American Geophysical Union, Washington, USA (2003)

Vincke, P.H.: Multicriteria decision-aid. John Wiley & Sons, Chichester (1992)

Visvanathan, A.,Youn, J.H., Deogun, J.: Hierarchical data dissemination scheme for large scale sensor networks. In: IEEE International Conference on Communications (ICC'05), pp. 3030–3036, May 2005

Viswanathan, N., Pick, J.B., Hettrick, W.J., Ellsworth, E.: An analysis of commonality in the twin metropolitan areas of San Diego, California and Tijuana, Mexico. J. Soc. Plann. Sci. **39**, 57–79 (2005)

Vitter, J.S.: External Memory Algorithms and Data Structures. ACM Comput. Surveys **33**(2), 209–271 (2001)

Vivid Solutions (2006): Java Conflation Suite (JCS) (2007) http://www.vividsolutions.com/JCS/. Accessed 20 Feb 2007

Vlachos, M., Hadjieleftheriou, M., Gunopulos, D., Keogh, E.: Indexing multi-dimensional time-series with support for multiple distance measures. In KDD 03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 216–225, Washington, D.C., USA, 24–27 Aug 2003

Vlachos, M., Kollios, G., Gunopulos, D.: Discovering similar multidimensional trajectories. In Proc. of IEEE ICDE, pp. 673–684 (2002)

Voges U, Senkler K (ed) (2004) OpenGIS Catalogue Services Specification 2.0 – ISO19115/ISO19119 Application Profile for CSW 2.0 OpenGIS Project Document 04-038r2. https://portal.opengeospatial.org/files/?artifact_id=8305

Vogt, H.: Pervasive Computing: First International Conference Proceedings, In: Efficient Object Identification with Passive RFID Tags, Lecture Notes in Computer Science, vol. 2414, pp. 98. Springer (2002)

Vretanos PA (ed) (2002) OpenGIS Filter Encoding Specification Version 1.1.0. OpenGIS Project Document 04-095. http://portal.opengeospatial.org/files/?artifact_id=8340

Vretanos PA (ed) OpenGIS Web Feature Service Implementation Specification Version 1.1.0. OpenGIS Project Document 04-094. https://portal.opengeospatial.org/files/?artifact_id=8339

Vyas, V.M., Tong, S.N., Uchrin, C., Georgopoulos, P.G., Carter, G.P.: Geostatistical estimation of horizontal hydraulic conductivity for the Kirkwood-Cohansey aquifer. J. Am. Water Res. Assoc. **40**, 187–195 (2004)

W3C OWL Ontology Language Reference. http://www.w3.org/TR/owl-ref

W3C: Scalable Vector Graphics (SVG) – XML Graphics for the Web (1999–2006), http://www.w3.org/Graphics/SVG/, accessed 2006–11–12

W3C Semantic Web. http://www.w3.org/2001/sw/. Accessed 21 March 2007

W3C: SVG Print Specification (2003–2006), http://www.w3.org/TR/SVGPrint/, accessed 2006–11–12

W3C (World Wide Web Consortium), SOAP Version 1.2, http://www.w3.org/TR/soap

Wackernagel, H.: Multivariate Geostatistics: An introduction with Applications, 3nd edn. Springer-Verlag, New York (2006)

Wackernagel, H.: Multivariate Geostatistics: An introduction with applications, 3rd edn., pp. 45–58. Springer-Verlag, New York, LLC (2003)

Wagner, W., Ullrich, A., Ducic, V., Melzer, T., Studnicka, N.: Gaussian decomposition and calibration of a novel small-footprint full-waveform digitising airborne laser scanner. ISPRS J. Photogramm. Remote Sensing **60**, 100–112 (2006)

Wakefield, J.: A critique of statistical aspects of ecological studies in spatial epidemiology. Environ. Ecol. Stat. **11**, 31–54 (2004)

Walid G.A., Ilyas, I.F.: A Framework for Supporting the Class of Space Partitioning Trees. TR01-002. Purdue University, West Lafayette, IN (2001)

Walker, A., Pham, B., Maeder, A.: A Bayesian framework for automated dataset retrieval. In: 10th International Multimedia Modelling Conference, Geographic Information Systems, pp. 138. (2004)

Walker, A., Pham, B., Moody, M.: Spatial Bayesian learning algorithms for geographic information retrieval. In: Proceedings 13th annual ACM international workshop on Geographic information systems, pp. 105–114. Bremen, Germany (2005)

Wallace, T., Wintz, P.: An efficient three-dimensional aircraft recognition algorithm using normalized fourier descriptors. Computer Graphics and Image Processing **13**, 99–126 (1980)

Waller, L.A., Gotway, C.A. Applied spatial statistics for public health data. Wiley, Hoboken, N.J. (2004)

Waller, L.: Methods for detecting disease clustering in time or space. In: Brookmeyer, R., Stroup, D. (eds.) Monitoring the Health of Populations: Statistical Methods and Principles in Public Health Surveillance, pp. 167–202. Oxford University Press, New York (2004)

Waller, L.A., Smith, D., Real, L.A., Childs, J.: Monte Carlo assessments of goodness of fit for ecological simulation models. Ecol. Model. **164**, 49–63 (2003)

Waller, L.A., Xia, B.P., Gelfand, A.E.: Hierarchical spatio-temporal mapping of disease rates. J. Am. Stat. Assoc. **92**, 607–617 (1997)

Walter, V., Fritsch, D.: Matching Spatial Data Sets: a Statistical Approach. Int. J. Geogr. Inf. Sci. **5**(1), 445–473 (1999)

Walton, C.A.: US Patent 3,752,960, 27 December 1971

Wang, B., Pan, F., Ren, D., Cui, Y., Ding, Q., Perrizo, W.: Efficient OLAP operations for spatial data using peano trees: In ACM Special Interest Group on Management of Data Workshop (SIGMOD), San Diego, California, 2003, pp. 28–34

Wang, G., Chen, H., Atabakhsh, H.: Automatically detecting deceptive criminal identities. Commun. ACM **47**(3):70–76 (2004)

Wang, J., Hsu, W., Lee, M.L.: A framework for mining topological patterns in spatio-temporal databases. In: Proceedings of the 14th ACM international conference on Information and knowledge management, Full paper in IEEE Trans. KDE, 16(12), 2004. (2005)

Wang, J., Zhang, Q.: Applicability of a Gradient Profile Algorithm for Road Network Extraction – Sensor Resolution and Background Consideration. Canadian J. Remote Sensing **26**(5), 428–439 (2000)

Wang, J.Z., Li, J.:System for screening objectionable images using daubechies' wavelets and color histograms. Interactive Distributed Multimedia Systems and Telecommunication Services, Proceedings of the Fourth European Workshop (IDMS'97) (1997)

Wang, L., Birol, I., Hatzimanikatis, V.: Metabolic control analysis under uncertainty: framework development and case studies. Biophysical Journal. **87**(6) (2004)

Wang, L., Wang, C., Xie, X., Forman, J., Lu, Y., Ma, W., Li, Y.: Detecting Dominant Locations from Search Queries. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005) (2005)

Wang, Y., Zheng, T.: Comparison of Light Detection and Ranging and National Elevation Dataset Digital Elevation Model on Floodplains of North Carolina. Nat. Hazards Rev. **6**(1), 34–40 (2005)

Want, R., Fishkin, K.P., Gujar, A., Harrison, B.L.: Bridging physical and virtual worlds with electronic tags, In: CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, New York (1999)

Want, R, Hopper, A., Falcão, V., Gibbons, J.: The Active Badge Location System. ACM Trans. Inf. Syst. **10**, pp. 91–102 (1992)

Want, R., Schilit, B.: Expanding the horizons of location-aware computing. IEEE Computer **34**(8), 31–34 (2001)

Want, R., Schilit, B.N., Adams, N.I., Gold, R., Petersen, K., Goldberg, D., Ellis, J.R., Weiser, M.: An overview of the parctab ubiquitous computing experiment. IEEE Personal Communications **2**(6), 28–43 (1995)

Ward, A., Jones, A., Hopper, A.: A new location technique for the active office. IEEE Personal Commun. Mag. 4(5), 42–47

Wardrop, D.H.: Digital tools for development, testing, and application of ecological indicators and early warning. In: Workshop on Digital Governance and Hotspot GeoInformatics at the Digital Government Research Conference, San Diego, CA, 21–24 May 2006

Ware, C.: Information Visualization. Morgan Kaufman, San Francisco (2000)

Ware, J.M., Jones, C.B.: Matching and Aligning Features in Overlayed Coverages. In: Proceedings of the 6th ACM Symposium on Geographic Information Systems. Washingnton, D.C. (1998)

Warner, K.: Geospatial Semantics: Why, of What, and How?. J. Data Semant., Lect. Notes Comp. Sci. **3534**, 1–24 (2005)

Warren, W.: Constructing an Econiche. In: Flack, J., et al. (eds.) Global Perspectives on the Ecology of Human-Machine Systems, p. 210–237. Lawrence Erlbaum Associates, Hillsdale, New Jersey (1995)

Warrior, J., McHenry, E., McGee, K.: They know where you are. Spectrum, **40**(7):20–25, July 2003

Warwick, J.J., Haness, S.J.: Efficacy of ARC/INFO GIS Application to Hydrologic Modeling, J. Water Resour. Plann Manag. **120**(3), 366–381 (1994)

Watson P.: Topology and ORDBMS Technology, a White Paper, 2002. Available at http://www.laser-scan.com/pdf/topo_ordbms.pdf

Wayne, L., Federal Geographic Data Committee: Institutionalize Metadata Before It Institutionalizes You. http://www.fgdc.gov/metadata/metadata-publications-list (2005)

Wayne, L.: Metadata In Action: expanding the utility of geospatial metadata. http://www.fgdc.gov/metadata/metadata-publications-list(2005)

WCRP – World Climate Research Programme. http://wcrp.wmo.int/. Accessed November 2006

Weber, R., Schek, H., Blott, S.: A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In: Proceedings of International Conference on Very Large Data Bases (1998)

Web Mapping Illustrated by Tyler Mitchell (O'Reilly Series, 2005). This book explains how to find, collect, understand, use, and share mapping data, both over the traditional Web and using OGC-standard services like WFS and WMS

Web Ontology Language http://www.w3.org/2004/OWL/

Web Services Activity, W3C: http://www.w3.org/2002/ws/

Webster, R., Oliver, M.A.: Geostatistics for Environmental Scientists. John Wiley and Sons, New York (2001)

Weibel, R., Dutton, G.: Generalizing Spatial Data and Dealing with Multiple Representations. In: Longley, P., Goodchild, M.F., Maguire, D.J., Rhind, D.W. (eds.) Geographical Information Systems, pp. 125–156. John Wiley, New York (1999)

Weibel, R., Dutton, G.: Generalizing spatial data and dealing with multiple representations. In: Longley, P.A., Goodchild, M.F., Maguire, D.J., Rhind, D.W. (eds.) Geographical Information Systems: Principles, Techniques, Management and Applications, 2nd edn. vol. 1, pp. 125–155. Wiley, New York (1999)

Wei, H., Chen, D.: The effect of spatial auto-correlation on the sampling design in accuracy assessment: A case study with simulated data. Environ. Inform. Arch. **2**, 910–919 (2004)

Weisburd, D., Eck, J.E. (eds.): Crime and Place: Crime Prevention Studies, vol. 4. Police Executive Research Forum: Willow Tree Press (1995)

Weisburd, D., Green, L.: Policing Drug Hot Spots: The Jersey City Drug Market Analysis Experiment. Justice Q. **12**(4), 711–736 (1995)

Weisburd, D., McEwen, T. (eds.): Introduction: Crime Mapping and Crime Prevention. In: Crime Mapping and Crime Prevention, pp. 1–23. Criminal Justice Press, Monsey, NY (1997)

Weiser, M.:. The computer for the 21st century. *Sci. Am.* **265**, 94–104 (1991), July 12th, 2007

Weiser, M.: Some computer science issues in ubiquitous computing. Commun. ACM **36**, 74–84 (1993)

Weisman, J.: Evaluating architectural legibility: Way-finding in the built environment. Environ. Behav. **13**, 189–204 (1981)

Weiss, M.J.: The clustered world: how we live, what we buy, and what it all means about who we are. 1st edn. Little, Brown, Boston, viii, 323 p (2000)

Weiss, M.J.: The clustering of America, 1st edn. Harper & Row, New York, xvi, 416 p (1988)

Weiss, M.J.: Latitudes & attitudes: an atlas of American tastes, trends, politics, and passions: from Abilene, Texas to Zanesville, Ohio. 1st edn. Little, Brown, Boston, 224 p (1994)

Weis, S.A., Sarma, S.E., Rivest, R.L., Engels, D.W.: Security in Pervasive Computing, In: Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems, Lecture Notes in Computer Science, vol. 2802, pp. 201–212. Springer (2004)

Wei, Y., Di, L., Zhao, B., Liao, G., Chen, A.: Transformation of HDF-EOS metadata from the ECS model to ISO 19115-based XML. Comput. Geosci. **33**(2):238–247 (2007)

Wen, Z., Zhou, M., Aggarwal, V.: An optimization-based approach to dynamic visual context management. In: InfoVis'05, pp. 111–118 (2005)

Werb, J., Lanzl, C.: Designing a positioning system for finding things and people indoors. IEEE Spectrum **35**, 71–78 (1998)

Whiteaker, T.L., Robayo, O., Maidment, D.R., Obenour, D. From NEXRAD rainfall map to an inundation map. J. Hydrol. Eng. **11**(1), 37–45 (2006)

White, D., Jain, R.,: Similarity indexing with the SS-tree. ICDE 516–523 (1996)

White, D.A., Jain, R.: Similary Indexing Algorithms and Performance. In: Proc. SPIE. 2670, pp. 62–73 (1996)

Whitehouse, K., Culler, D.: Calibration as parameter estimation in sensor networks. In: WSNA '02: Proc. ACM Int. workshop on Wireless sensor networks and applications. (2002)

White, M.: The Theory of Geographical Data Conflation. Internal Census Bureau draft document (1981)

Whittaker, G.: Application of SWAT in the evaluation of salmon habitat remediation policy. Hydrol. Process. **19**(3), 839–848 (2005)

Whittle, P.: On Stationary Processes in the Plane. Biom., **41**(3), 434–449 (1954)

Wickham, J.D., Stehman, S.V., Smith, J.H., Yang, L.: Thematic accuracy of the 1992 National Land-Cover Data for the western United States. R. Sens. Environ. **91**, 452–468 (2004)

Widom, J.: Trio: A system for integrated management of data, accuracy, and lineage. In: Proc. CIDR, 2005

Wikipedia (2006) Global Positioning System. http://en.wikipedia.org. Accessed 8 May 2006

Wikipedia (2006) Trilateration. http://en.wikipedia.org. Accessed 8 May 2006

Wikipedia http://en.wikipedia.org/wiki/Geographic_information_system Accessed 23 March 2007

Wikipedia. Image compression. http://en.wikipedia.org/wiki/Image_compression

Wikipedia: Radio Frequency Identification. http://en.wikipedia.org/w/index.php?title=Radio_Frequency_Identification&oldid=85251329, Wikipedia, The Free Encyclopedia: Retreived 10/10/2006 (2006)

Wikipedia, Traveling Salesman Problem. http://en.wikipedia.org/

Wikipedia – Webmapping. History of web mapping, http://en.wikipedia.org/wiki/Web_mapping#History_of_web_mapping (2007) accessed 2007-01-05

Wikle, C.: Hierarchical models in environmental science. Int. Stat. Rev. **71**, 181–199 (2003)

Wikle, C.K., Berliner, L.M., Cressie, N.: Hierarchical Bayesian space-time models. J. Envr. Ecol. Stat. **5**, 117–154 (1998)

Wikle, C.K., Cressie, N.: A dimension-reduced approach to space-time Kalman filtering. Biom. **86**(4), 815–829 (1999)

Wikle, C.K.: Hierarchical Bayesian models for predicting the spread of ecological processes. Ecol. **84**, 1382–1394 (2003)

Wikle, C.K., Milliff, R.F., Nychka, D., Berliner, L.M.: Spatiotemporal hierarchical Bayesian modeling: Tropical ocean surface winds. J. Am. Stat. Assoc. **96**, 382–397 (2001)

Wilensky, U.: Netlogo flocking model. http://ccl.northwestern.edu/netlogo/models/Flocking

Williams, B., Tagliaferri, P.A., Meinhold, S.S., Hummer, J.E., Rouphail, N.M.: Simulation and Analysis of Freeway Lane Reversal for Coastal Hurricane Evacuation, ASCE J. Urban Plann. Dev. – Special Emergency Transportation Issue, **133**(1), 61–72 (2007)

Williams, C.K.I., Barber, D.: Bayesian Classification with Gaussian Processes. IEEE PAMI **20**(12), 1342–1351 (1998)

Williams, C.K.I.: Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond. In: Jordan, M.I.(ed.) Learning in Graphical Models, pp. 599–621. MIT Press, Cambridge, MA (1998)

Williams, D.: The Definitive Guide To GPS, RFID, Wi-Fi, and Other Wireless Location-Based Services – What Applications Will Succeed in What Markets, and How To Design and Market Them (2005)

Williams, J.: Yosemite Hiking Map 2.0 (2006), http://www.carto.net/williams/yosemite/, accessed 2006–11–12

Williamson, O.E.: The new institutional economics: taking stock, looking ahead. J. Econ. Literat. **38**, 595–613 (2000)

Wilson, J.P. (ed.), Gallant, J.C.: Terrain Analysis: Principles and Applications, p. 512, ISBN 0-471-32188-5. John Wiley & Sons, New Jersey, USA (2000)

Wilson, J.Q., Kelling, G.: Broken Windows: The Police and Neighborhood Safety. Atlantic Monthly **29**(3), 29–38 (1982)

Wilson, R.J., Beineke, L.W. (eds.): Applications of graph theory. Academic Press, London, New York (1979)

Wilson, R.J.: Introduction to Graph Theory. Longman Inc., New York (1985)

Wilson, R.: Tennessee Geographic Information Council, Introduction to Metadata (2004)

Wilson, R.:What is metadata? TNGIC Metadata Outreach Trainers and Power Point Modules. http://www.tnmetadata.org/training.html (2004)

Winter, S.: Modeling Costs of Turns in Route Planning. GeoInformatica **6**(4), 363–380 (2002)

Witten, I.H., Wyvill, B.: On the generation and use of space-filling curves. Software – Practice and Experience **3**, 519–525 (1983)

Wolf, P.R., Dewitt, B.A.: Elements of Photogrammetry with Applications in GIS, McGraw-Hill, New York (2000)

Wolfson, O., Chamberlain, S., Dao, S., Jiang, L., Mendez, G.: Cost and imprecision in modeling the position of moving objects. Proc. 14th Int. Conf. on Data Engineering, Orlando, Florida, pp. 588–596 (1998)

Wolfson, O.: Moving objects information management: The database challenge. In: NGITS '02: Proceedings of the 5th International Workshop on Next Generation Information Technologies and Systems, pp. 75–89. Springer, London (2002)

Wolfson, O., Sistla, A.P., Chamberlain, S., Yesha, Y.: Updating and querying databases that track mobile units. Distrib. Parallel Database **7**, 257–387 (1999)

Wolfson, O., Xu, B., Chamberlain, S., Jiang, L.: Moving Objects Databases: Issues and Solutions. In: Proceedings of the 10th Int. Conf. on Scientfic and Statistical Database Management, Capri, Italy, 1–3 July 1998, pp. 111–122

Wolfson, O., Xu, B., Yin, H.B., Cao, H.: Search-and-discover in mobile P2P network databases. In Proc. of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS'06), Lisbon, Portugal (2006)

Wolshon B., Catarella-Michel, A., Lambert, L.: Louisiana Highway Evacuation Plan for Hurricane Katrina: Proactive Management of Regional Evacuations. ASCE J. Transp. Eng. **132**(1), 1–10 (2006)

Wolshon, B., Lambert, L.: Convertible Lanes and Roadways, National Cooperative Highway Research Program, Synthesis 340, Transportation Research Board, p. 92. National Research Council, Washington, DC (2004)

Wolshon B. and Marchive, E.: Evacuation Planning in the Urban-Wildland Interface: Moving Residential Subdivision Traffic During Wildfires. ASCE J. Urban Plann. Dev. – Special Emergency Transportation Issue **133**(1), 73–81 (2007)

Wolshon B. and McArdle, B.: Temporospatial Analysis of Hurricane Katrina Regional Evacuation Traffic Patterns, ASCE J. Infrastruct. Syst. – Special Infrastructure Planning, Design, and Management for Big Events Issue (in press)

Wolshon, B.: One-Way-Out: Contraflow Freeway Operation for Hurricane Evacuation. Nat. Hazard. Rev., Am. Soc. Civil Eng. **2**(3), 105–112 (2001)

Wolshon, B.: Planning and Engineering for the Katrina Evacuation. The Bridge, National Acad. Sci. Eng., vol. 36, no. 1, pp. 27–34 (2006)

Wolshon, B., Urbina, E., Levitan, M.: National review of hurricane evacuation plans and policies. In: Technical report, Hurricane Center. Louisiana State University, Baton Rouge, Louisiana (2002)

Wolshon, B., Urbina, E., Levitan, M., Wilmot, C.: National Review of Hurricane Evacuation Plans and Policies, Part II: Transportation Management and Operations; ASCE Nat. Hazards Rev. **6**(3), 142–161 (2005)

Woodcock, C.E., Gopal, S.: Fuzzy set theory and thematic maps: accuracy assessment and area estimation. Int. J. Geogr. Inf. Sci. **14**(2), 153–172 (2000)

Woodruff, A.G., Plaunt, C.: GIPSY: geo-referenced information processing system. J. Am. Soc. Inf. Sci. **45**, 645–655 (1994)

Woods, D.D.: Visual momentum: A concept to improve the cognitive coupling of person and computer. Int. J. Man-Machine Stud. **21**, 229–44 (1984)

Woo, A., Tong, T., Culler, D.: Taming the underlying challenges of reliable multihop routing in sensor networks. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (ACMSensys), pp. 14–27. Los Angeles, November 2003

Worboys, M., Clementini, E.: Integration of imperfect spatial information. J Vis Language Comput **12**, 61–80 (2001)

Worboys, M.F.: Computation with imprecise geospatial data. Computers, Environment and Urban Systems **22**, 85–106 (1998)

Worboys, M.F., Duckham, M.: Monitoring qualitative spatiotemporal change for geosensor networks. Int. J. Geogr. Inf. Sci. **20**(10), 1087–1108 (2006)

Worboys, M.F.: Event-oriented approaches to geographic phenomena. Int. J. Geogr. Inf. Sci. **19**(1), 1–28 (2005)

Worboys, M.F.: Imprecision in finite resolution spatial data. GeoInformatica **2**, 257–279 (1998)

Worboys, M.F.: Nearness relations in environmental space. Int. J. Geogr. Inf. Sci. **15**(7), 633–651 (2001)

Worboys, M.F.: A Unified Model for Spatial and Temporal Information. In: Computer Journal, Vol.37, No.1, pp. 26–34. Oxford University Press, Oxford, UK (1994)

World Commission on Environment and Development: Our Common Future. Oxford University Press (1987)

World Wildlife Fund: The Living Planet Report 2006. WWF, Switzerland (2006)

Wortman, K., Buttenfield, B.: Special Issue. Current Developments and Use of the Spatial Data Transfer Standard. Cartography Geogr. Inf. Syst., **21**(3), entire issue (1994)

Wright, J.: A method of mapping densities of population: with Cape Cod as an example. Geogr. Rev. **26**, 103–110 (1936)

Wright, R.: Private eyes. New York Times Mag. September 5, 1999, Final Section 6, Page 50, Column 1.

Writing of aspects of this paper was funded and supported by the USDA (Cooperative State Research, Education, and Extension Service) Rural GIS program (see www.ruralgis.org) and EPA's Office of Environmental Information

Wu, Y., Agrawal, D., El Abbadi, A.: A comparison of DFT and DWT based similarity search in time-series databases. In: Proceedings of the 9th ACM CIKM International Conference on Information and Knowledge Management. McLean, 6–11 November 2000, pp. 488–495

Wu, Y-H. and Miller, H.J.: Computational tools for measuring space-time accessibility within transportation networks with dynamic flow. J. Transp. Stat. **4**(2/3), 1–14 (2002)

Wu, Y.: Raster, vector, and automated raster-to-vector conversion. In: Kenney, A.R., Rieger, O.Y. (eds.) Moving Theory into Practice:

Digital Imaging for Libraries and Archives, Research Libraries Group (2000)

Wynne, D.L., Klein, L., Gejjagaraguppe, R., Ilg, D. (1997) HDF-EOS Library User's Guide for the ECS Project, Volume 1: Overview and Examples. Hughes Applied Information Systems. doi:170-TP-005-004

Xiang, Z., Song, S., Chen, J., Wang, H., Huang, J., Gao, X.: A wireless LAN-based indoor positioning technology. IBM J. Res. Dev. **48**, 617–626 (2004), Armonk, New York, USA

Xia, T., Zhang, D.: Continuous reverse nearest neighbor monitoring. In: ICDE '06: Proceedings of the 22nd International Conference on Data Engineering (ICDE '06), p. 77. Washington, DC, USA (2006)

Xie, J., Li, Z., Xu, T. (2005) Oracle Database 10*g* GeoRaster: Scalability and Performance Analysis. http://www.oracle.com/technology/products/spatial/index.html (2007). Accessed 1 February 2007

Xie, J., Li, Z., Xu, W.: Using Enterprise Grid Computing Technologies to Manage Large-Scale Geoimage And Raster Databases. In: The Proceedings of ASPRS 2006 Annual Conference, Reno, Nevada, May 1–5, 2006

Xie, J., Sharma, J. (2005) Oracle Database 10*g*, Managing Spatial Raster Data Using GeoRaster—Technical White Paper. http://www.oracle.com/technology/products/spatial/index.html (2007). Accessed 1 February 2007

Xie, L., Bao, S., Pietrafesa, L., Foley, K., Fuentes, M.: A real-time hurricane surface wind forecasting model: Formulation and verification. Monthly Weather Review **134**, 1355–1370 (2006)

Xiong, H., Shekhar, S., Huang, Y., Kumar, V., Ma, X., Yoo, J.S.: A framework for discovering co-location patterns in data sets with extended spatial objects. In: Proceedings of the Fourth SIAM International Conference on Data Mining (SDM04), Lake Buena Vista, Florida, (2004)

Xiong, X., Aref, W.G.: R-trees with Update Memos. In: Proceedings of ICDE, p. 22. Atlanta, Georgia, USA, 3–8 Apr 2006

Xiong, X.; Mokbel, M.F. Aref, W.G.: SEA-CNN: Scalable processing of continuous K-nearest neighbor queries in spatio-temporal databases. In: ICDE (2005)

Xi, X., Keogh, E.J., Shelton, C.R., Li, W., Ratanamahatana, C.A.: Fast time series classification using numerosity reduction. In: Cohen, W.W., Moore, A. (eds.): Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, 25–29 June 2006, pp. 1033–1040. ACM 2006

XQuery 1.0: An XML Query Language, W3C Working Draft. http://www.w3.org/TR/xquery/. Accessed 3 Dec 2006

Xu, B., Wolfson, O.: Data Management in Mobile P2P Networks, Springer Verlag Lecture Notes in Computer Science, Proc. of the 2nd International Workshop on Databases, Information Systems, and P2P Computing (DBISP2P'04), Toronto, Canada, Aug 2004

Xu J., Chen, H.: Untangling criminal networks: A case study. In: ISI, pp. 232–248 (2003)

Xu, Y., Heidemann, J., Estrin, D.: Geography-informed energy conservation for ad hoc routing. In: The ACM International Conference on Mobile Computing and Networking, pp. 70–84, Rome, Italy, July 2001

Yanai, K.: Web image mining toward generic image recognition. In: Proceedings of the Twelfth International World Wide Web Conference (2003)

Yang, B.: A multi-resolution model of vector map data for rapid transmission over the internet. Computers & Geosciences **31**(5), 569–578 (2005)

Yang, B., Purves, R., Weibel, R.: Efficient transmission of vector data over the internet. Int. J. Geogr. Inf. Sci. **21**, 215–237 (2007)

Yang C., Evans J., Bambacus M., Cole M., Alameh N., Marley S.: Spatial Web Portal: Architecture and Example. PE&RS 73(6):691–698 (2006)

Yang, C., Tao, V.: Distributed geospatial information service. In: Rana, S., Sharma, J. (eds.) Frontiers of Geographic Information Technology, chap. 5, pp. 113–130. Springer, London (2005)

Yang, C., Tao, V.: Distributed geospatial information service In: Rana, S., Sharma, J. (eds.) Frontiers of Geographic Information Technology, pp. 113–130. Springer, London (2005)

Yang, C., Wong, D., Kafatos, M., Yang, R.: Implementing computing techniques to accelerate network GIS. In: Li, D., Xia, L. (eds.) Proceedings of the SPIE Vol. 6418, 64181C, Geoinformatics (2006): GNSS and Integrated Geospatial Applications, Wuhan, China, 27–29 Oct 2006

Yang, C., Wong, D., Kafatos, M., Yang, R.: Network GIS. In: Qu, J., Gao, Y., Kafatos, M., Murphy, R., Solomonson, V. (eds.) Earth Science Satellite Remote Sensing, vol. 2, pp. 252–269. Springer and Tsinghua University Press, Beijing (2006)

Yang, C., Wong, D., Li, B.: Computing and computational issues of distributed GIS. Int. J. Geogr. Inf. Sci. **10**(1), 1–3 (2005)

Yang, C., Wong, D., Yang, R., Kafatos, M., Li, Q.: WebGIS performance improving techniques. Int. J. Geogr. Inf. Sci. **19**, 319–342 (2005)

Yang, H., Parthasarathy, S., Mehta, S.: A generalized framework for mining spatio-temporal patterns in scientific data. In KDD 2005: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pp. 716–721. ACM Press, New York, NY, USA (2005)

Yang, H., Parthasarathy, S., Ucar, D.: A spatio-temporal mining approach towards summarizing and analyzing protein folding trajectories. Algorithms Mol. Biol. **2**(3) (2007)

Yang, H.: Srinivasan Parthasarathy, and Sameep Mehta. Mining spatial object associations for scientific data. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (2005)

Yao, X.: Research Issues in Spatio-temporal Data Mining. A White Paper submitted to the University Consortium for Geographic Information Science (UCGIS) workshop on Geospatial Visualization and Knowledge Discovery, Lansdowne, Virginia, 18–20 Nov 2003

Yao, Y., Gehrke, J.E.: The cougar approach to in-network query processing in sensor networks: ACM Special Interest Group on Data Management of Data (SIGMOD) Record, pp. 9–18 (2002)

Yedavalli, K., Krishnamachari, B., Ravula, S., Srinivasan, B.: Ecolocation: A sequence based technique for RF-only localization in wireless sensor networks. In: Proc. 4th Int. Conf. on Information Processing in Sensor Networks (IPSN '05), April (2005)

Ye, N.: Handbook of Data Mining. Laurence Earlbaum, Mahwah NJ (2003)

Yi, B., Faloutsos, C.: Fast time sequence indexing for arbitrary lp norms. In: Proceedings of the 26th International Conference on Very Large Databases, Cairo, Egypt, 10–14 September 2000, pp. 385–394

Yiu, M.L., Mamoulis, N, Papadias, D.: Aggregate Nearest Neighbor Queries in Road Networks. TKDE **17**(6), 820–833 (2005)

Yiu, M.L., Papadias, D., Mamoulis, N., Tao, Y.: Reverse Nearest Neighbors in Large Graphs. TKDE **18**(4), 540–553 (2006)

Yiu, M.L., Tao, Y., Mamoulis, N.: The B^{dual}-Tree: Indexing Moving Objects by Space-Filling Curves in the Dual Space. VLDB J., To Appear

Yiu, M., Mamoulis, N.: Clustering Objects on a Spatial Network. In: Proc. of SIGMOD (2004)

Yoder, R.C.: Design Consideration for Implementing Complex Client/Server Applications: a Case Study of Octree-Based 3-D GIS. (1996)http://www.albany.edu/~sysrcy/dissertation/PR-11.TXT. Accessed 8 April 2006

Young, M.I.: Inclusion and Democracy. Oxford Political Theory. Oxford University Press, Oxford, New York (2000)

Yousief, M.: Horus: A WLAN-based indoor location determination system, Ph.D. dissertation, University of Maryland (2004)

Youssef, M., Agrawala, A.: The Horus WLAN Location Determination System. In: Third International Conference on Mobile Systems, Applications, and Services (MobiSys 2005), June (2005)

Youssef, M., Atluri, V., Adam, N.R.: Preserving mobile customer privacy: an access control system for moving objects and customer profiles. In: MDM '05: Proceedings of the 6th international conference on Mobile Data Management, pp. 67–76, New York, NY, USA, ACM Press (2005)

Yuan, M., McIntosh, J.: A typology of spatiotemporal information queries. In: Shaw, K., Ladner, R., Abdelguerfi, M. (eds.) Mining Spatiotemporal Information Systems. Kluwer, Norwell, Massachusetts, USA (2002)

Yuan, M.: Representing complex geographic phenomena in GIS. Cartogr. Geogr. Inf. Sci. **28**, 83–96 (2001)

Yuan, M.: Use of a three-domain repesentation to enhance GIS support for complex spatiotemporal queries. Trans. GIS **3**, 137–159 (1999)

Yu, C., Ooi, B.C., Tan, K.-L., Jagadish, H.: Indexing the distance: an efficient method to KNN processing. In: Proceedings of International Conference on Very Large Data Bases (2001)

Yu, H-L, Chen, J.-C., Christakos, G., Jerrett, M.: Estimating residential level ambient PM10 and Ozone exposures at multiple timescales in the Carolinas with the BME method. In: Internal Report IR09, Geography Department, San Diego State University, San Diego, CA (2007)

Yu, H.: Spatio-temporal GIS design for exploring interactions of human activities. Cartography Geogr. Inf. Sci. **33**, 3–19 (2006)

Yu, Y., Govindan, R., Estrin, D.: Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks, Technical Report UCLA/CSD-TR-01-0023,UCLA, May 2001

Zabel, J.E.a.K.A.K.: Demand for air quality. Sage Urban Stud. Abstr. **29**(1), 3–135 (2001)

Zadeh, L.A.: Fuzzy Sets. Information and Control **8**, 338–353 (1965)

Zaiane, O.R., Han, J.W.: Mining multimedia data. CASCON'98: Meeting of Minds, pp 83–96, Toronto, Canada, November 1998

Zaki, M.J.: Efficiently mining frequent trees in a forest. In: Proc. of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2002)

Zaki, M.J., Gouda, K.: Fast vertical mining using diffsets. In: Proc. of ACM SIGKDD Conference (2003)

Zaki, M.: Scalable algorithms for association mining. IEEE Trans Knowl Data Eng **12**(3), 372–390 (2000)

Zaki, M.: SPADE: An efficient algorithm for mining frequent sequences. Machine Learning, **42**(1/2):31–60 (2001)

Zaninetti, L.: The galaxies distribution as given from the voronoi diagrams. In: Journal of Astronomy and Astrophysics, Italy

Zaslavsky, I., Baru, C., Bhatia, K., Memon, A., Velikhov, P., Veytser, V.: Grid-enabled mediation services for geospatial information. In: Agouris, P., Croitoru, A. (eds.) Next Generation Geospatial Information. From Digital Image Analysis to Spatiotemporal Databases, pp. 15–24. Taylor and Francis, London, UK (2005)

Zeiler, M.: Modeling our world: the ESRI guide to database design. ESRI, Redlands, California (1999)

Zeimpekis, V., Giaglis, G. M., Lekakos, G.: A taxonomy of indoor and outdoor positioning techniques for mobile location services. J. ACM Special Interest Group Electron. Commerce **3**, 19–27 (2003)

Zeinalipour-Yazti, D., Lin, S., Gunopulos, D.: Distributed spatio-temporal similarity search. In: CIKM 06: Proceedings of the 15th ACM international conference on Information and knowledge management, pp. 14–23, Arlington, VA, USA, 6–11 Nov 2006

Zeinalipour-Yazti, D., Vagena, Z., Gunopulos, D., Kalogeraki, V., Tsotras, V., Vlachos, M., Koudas, N., Srivastava, D.: The threshold join algorithm for top-k queries in distributed sensor networks. In: DMSN 05: Proceedings of the 2nd international workshop on Data management for sensor networks, pp. 61–66, Trondheim, Norway, 29 Aug 2005

Zerger, A., Smith, D.I.: Impediments to using GIS for real-time disaster decision support. Comput., Environ. Urban Syst. **27**, 123–141. (2003)

Zernike, F.: Beugungstheorie des Schneidenverfahrens und seiner verbesserten Form, der Phasenkontrastmethode. Physica, **1**, 689–704 (1934)

Zevenbergen, J., Frank, A., Stubkjær, E. (eds.): Real Property Transactions: Procedures, Transaction Costs and Models. IOS Press, Amsterdam (2007)

Zhang, B.S., et al.: Predictive modeling of hill-pasture productivity: integration of a decision tree and a geographical information system. Agri. Sys. **87**(1), 1–17 (2006)

Zhang, D., Tsotras, V.J., Gunopulos, D.: Efficient Aggregation over Objects with Extent. In: PODS, pp. 121–132 (2002)

Zhang, G., Parashar, M.: Dynamic context-aware access control for grid applications In: Proceedings of GRID, Phoenix, Arizona, November 17, 2003

Zhang, J., Hsu, W., Lee, M.L.: An information-driven framework for image mining. In: Proceedings of 12th International Conference on Database and Expert Systems Applications (DEXA), Munich, Germany (2001)

Zhang, J., Patel, V.: Distributed cognition, representation, and affordance. Pragmatics & Cognition **14**(2), 333–341 (2006)

Zhang, J.-X., Goodchild, M.F.: Uncertainty in Geographical Information. Taylor and Francis, New York (2002)

Zhang, J., Zhu, M., Papadias, D., Tao, Y., Lee, D.L.: Location-Based Spatial Queries. In: Proceedings of the 2003 ACM SIGMOD international conference on Management of data. pp. 443–454. ACM Press (2003)

Zhang, L., Li, Y., Rao, F., Yu, X., Chen, Y., Liu, D.: An approach to enabling spatial OLAP by aggregating on spatial hierarchy. In: Proc. 2003 Int. Conf. Data Warehousing and Knowledge Discovery (DaWaK'03), pp. 35–44. Prague, Czech Rep., Sept. (2003)

Zhang, P., Huang, Y., Shekhar, S., Kumar, V.: Correlation Analysis of Spatial Time Series Datasets: A Filter-and-Refine Approach. In: Lecture Notes in Computer Science, vol. 2637. Springer, Berlin/Heidelberg (2003)

Zhang, P., Huang, Y., Shekhar, S., Kumar, V.: Exploiting Spatial Autocorrelation to Efficiently Process Correlation-Based Similarity Queries. In: Lecture Notes in Computer Science, vol. 2750. Springer, Berlin/Heidelberg (2003)

Zhang, P., Steinbach, M., Kumar, V., Shekhar, S., Tan, P., Klooster, S., Potter, C.: Discovery of Patterns of Earth Science Data Using Data Mining. In: Next Generation of Data Mining Applications (2004)

Zhang, X., Mamoulis, N., Cheung, D.W.L., Shou, Y.: Fast mining of spatial collocations. In: Proc. of ACM SIGKDD Conference (2004)

Zhang, Y., Webber, R.E.: Space diffusion: An improved parallel halftoning technique using space-filling curves. In: Computer Graphics Proc., August 1993, pp. 305–312

Zhao, F., Guibas, L.: Wireless Sensor Networks: an Information Processing Approach. Morgan Kaufmann, San Francisco, CA (2004)

Zhao, J., Lu, C.T., Kou, Y.: Detecting region outliers in meteorological data. In: Proceedings of ACM International Symposium on Advances in Geographic Information System, New Orleans, Louisiana, 2003, pp.49–55

Zhao, J.L., Zaki, A.: Spatial Data Traversal in Road Map Databases: A Graph Indexing Approach. In: Proc. of Int. Conf. on Information and Knowledge Management (CIKM), Gaithersburg, MD, USA, 1994. ACM, pp. 355–362 (1994)

Zheng, B., Lee, D.: Semantic Caching in Location-Dependent Query Processing. SSTD, Redondo Beach, CA, USA, 12–15 July 2001

Zhong, Z.-N., Jing, N., Chen, L., Wu, Q.-Y.: Representing topological relationships among heterogeneous geometry-collection features. J. Comp. Sci. Tech. archive. Institute of Computing Technology, Beijing, China **19**(3), 280–289 (2004)

Zhou, M., Aggarwal, V.: An optimization-based approach to dynamic data content selection in intelligent multimedia interfaces. In: Proc. UIST '04, pp. 227–236 (2004)

Zhou, M., Chen, M.: Automated generation of graphic sketches by examples. In: IJCAI '03, pp. 65–71 (2003)

Zhou, M., et al.: Enabling context-sensitive information seeking. In: Intelligent User Interfaces, pp. 116–123 (2006)

Zhou, N.: A study on automatic ontology mapping of categorical information. In: Proceedings of National Conference on Digital Government Research, pp. 401–404. 18–21 May, Boston (2003)

Zhou, P., Zhang, D., Salzberg, B., Cooperman, G., Kollios, G.: Close pair queries in moving object databases. In: GIS '05: Proceedings of the 13th annual ACM international workshop on Geographic information systems, pp. 2–11. ACM Press, New York, NY, USA (2005)

Zhou, X., Truffet, D., Han, J.: Efficient polygon amalgamation methods for spatial OLAP and spatial data mining. In: Proc. 1999 Int. Symp. Large Spatial Databases (SSD'99), pp. 167–187. Hong Kong, China, July (1999)

Zhuang, V., Wrazien, D., Wang, M.H., Huang, X.W.: Programming ASP.NET for ArcGIS Server, Thomson Delmar Learing Inc., Clifton Park, NY (2005)

Zhu, H., Su, J., Ibarra, O.H.: Trajectory queries and octagons in moving object databases. In: CIKM, pp. 413–421 (2002)

Zhu, A.X., Band, L.E., Dutton, B., Nimlos, T.: Automated soil inference under fuzzy logic. Eco. Mod. **90**(2), 123–145 (1996)

Zhu, Y., Shasha, D.: Query by humming: a time series database approach. In Proc. of ACM SIGMOD (2003)

Ziegler, P., Klaus, R., Dittrich, K.:Three Decades of Data Integration – All Problems Solved?". Paper presented at the 18th World Computer Congress, Toulouse, France, 22–27 Aug 2004. http://arvo.ifi.unizh.ch/dbtg/Staff/Ziegler/papers/ZieglerWCC2004.pdf. Accessed 11 Sept 2007

Zienkiewics, O.C., Taylor, R.L.: Finite Element Method, Vol. 1, The Basis. Butterworth Heinemann, London (2000)

Zobel, J., Moffat, A.: Inverted Files for Text Search Engines. ACM Computing Surveys, **38**(2): 1–56 (2006)

Zubieta, J.C., Skinner, R., Dean, A.G.: Initiating informatics and gis support for a field investigation of bioterrorism: The new jersey anthrax experience. Int. J. Health Geograph. **2**(8). http://www.ij-healthgeographics.com/content/2/1/8 (2003)

Entry List